1996 / SZEPTEMBER

ÁRA: 356 FT

ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI FOLYÓIRAT LEMEZMELLÉKLETTEL



A HÓNAP TÉMÁJA:

SZOFTVERBÁBEL

A titokmegosztás

Grétsy a számítógépben

Leveles játékok varázsa

Kritikai szoftver-vitriológia

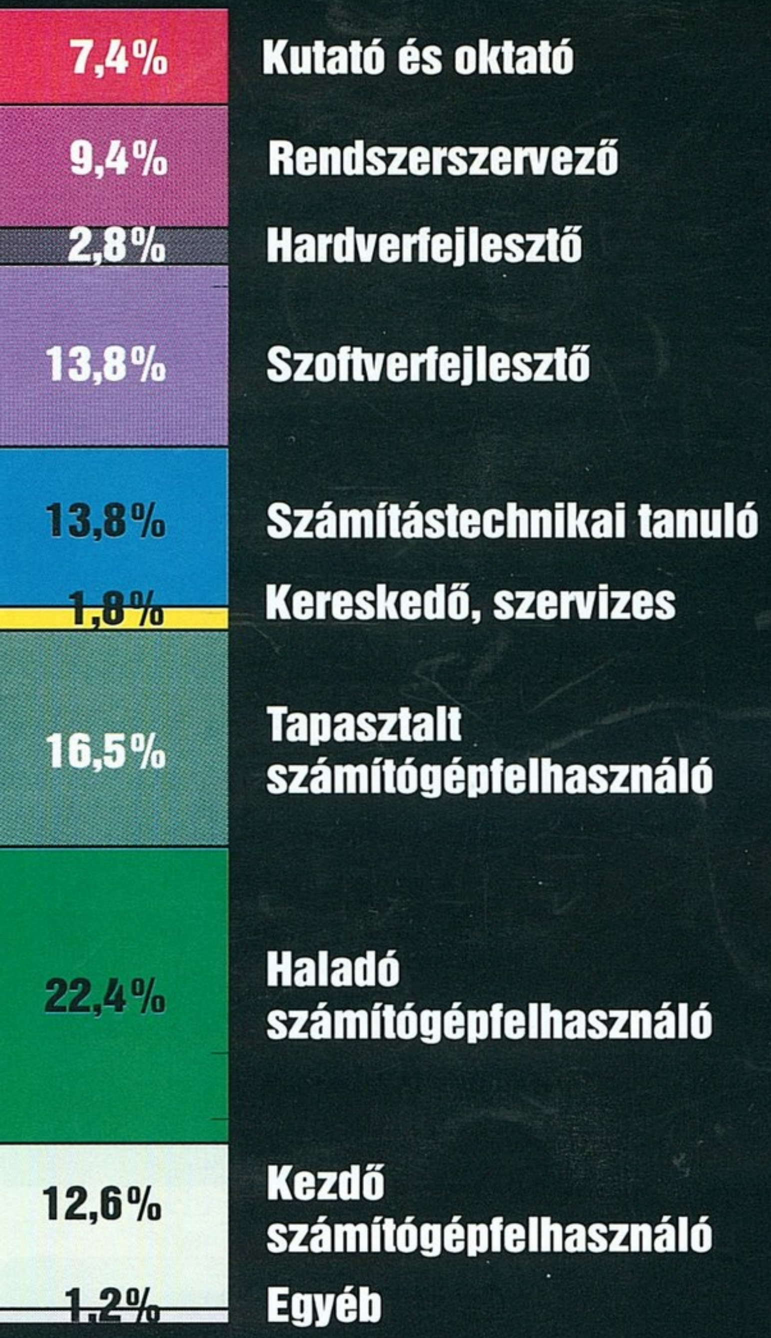
Szótár
Vegyprogramok
Elektronikus svábbogár
Mágnesezés-frissítő
Egérinfó
Körtetris

1996 / XIV. ÉVFOLYAM / MEGJELENIK HAVONTA / ÁRA: 356 FT

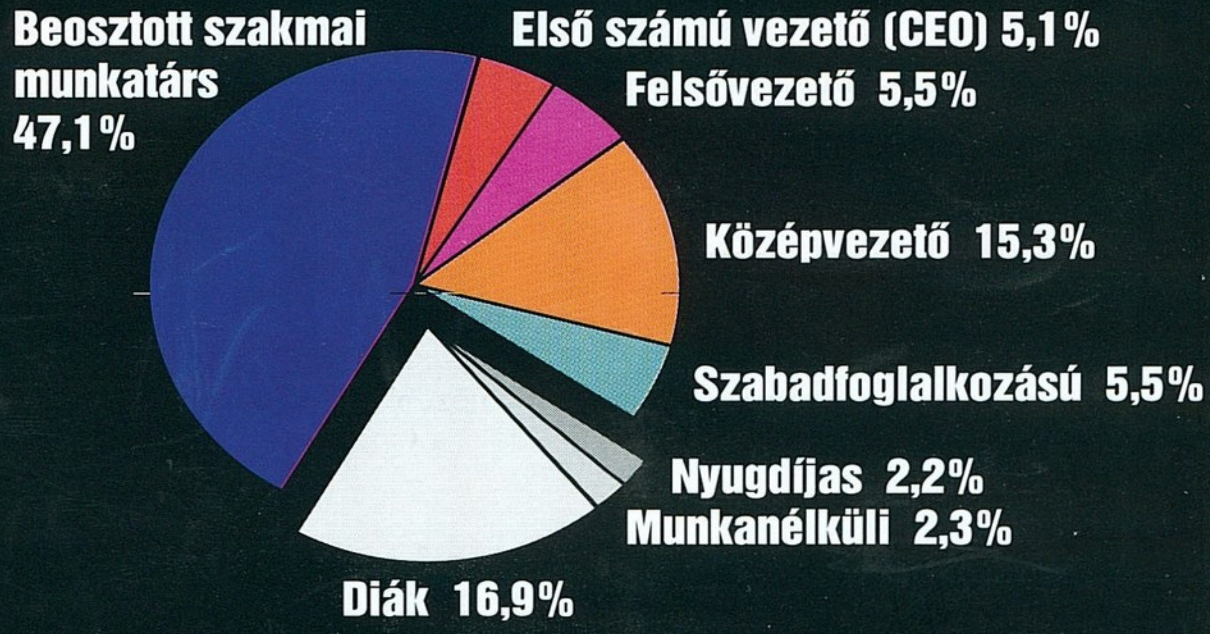
ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI FOLYÓIRAT LEMEZMELLÉKLETTEL

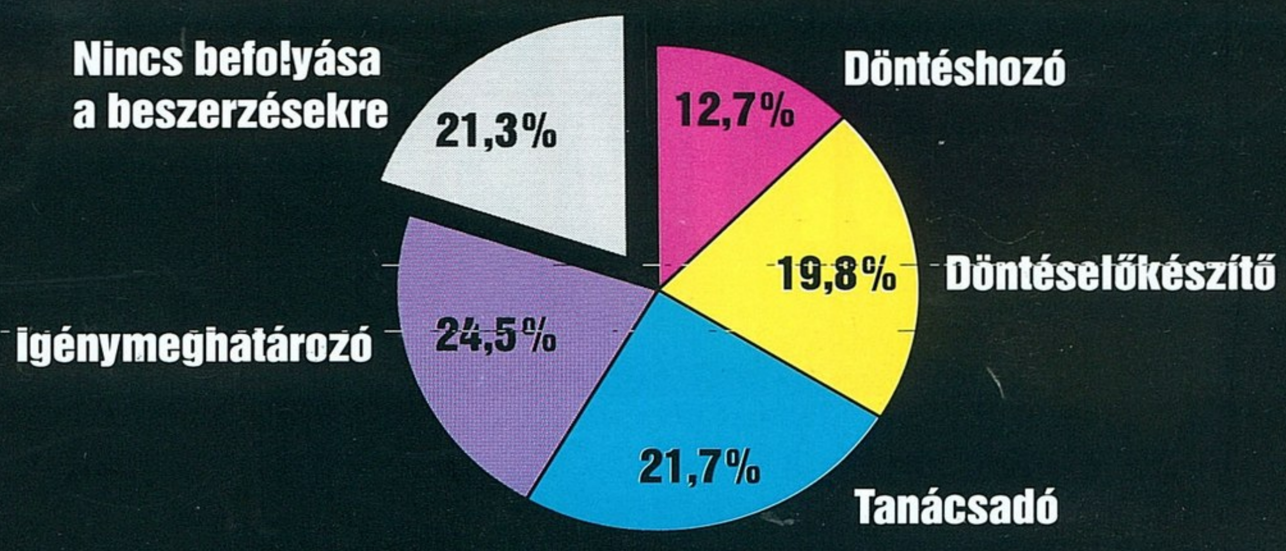
Szakmai tudásszint



Döntési kompetencia



Befolyás a munkahelyi vásárlásokra





TÉNYEK AZOKNAK, AKIK TÉNYEK ALAPJÁN (MÉDIA)TERVEZNEK



ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató számítástechnikai folyóirat

Megjelenik havonta, mágneslemez melléklettel

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Aszalós László, Csórián Sándor, Feleki Zoltán, Ferenczi Gábor, Herczeg József, Horlai János, Kis János, Nagy Gábor, Pogány Csaba, Szondi Egon János, Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:
1539 Budapest I., Márvány u. 17.
Telefon: 156-3211 / 200, 214
Fax (manuális): 156-3211 / 201
E-mail: alaplap@mail.datanet.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
Ez a szám 10 400 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Somogyi Tibor ügyvezető igazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft, számos számítástechnikai szaküzlet és más alternatív terjesztő

Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11701004-20171649
Eladási ár: 356 Ft
Évi előfizetési díj: 3564 Ft

Külföldre terjeszti a Kultúra,
H-1389 Budapest, Pf. 149

HU ISSN 1217-7598

Magyarul vagy nem magyarul

A számítástechnikával kapcsolatos magyar nyelvhasználat kérdésében nagyon megosztott a szakma, de a nyelvészek szűk, és a felhasználók széles tábora is.

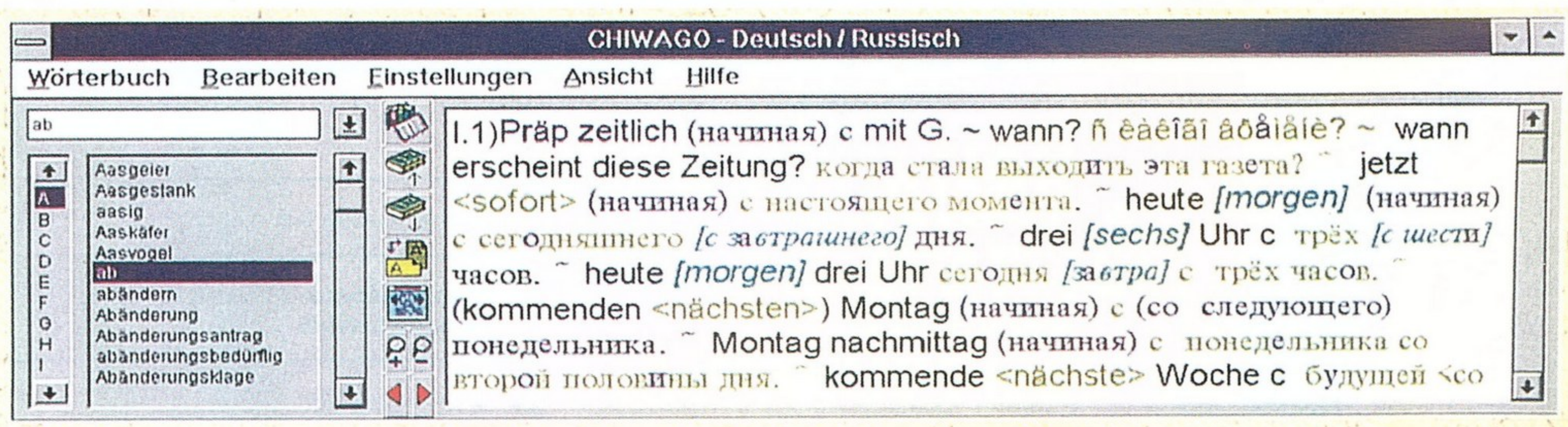
Az egyetértés abban még megvan, hogy a számítógéppel készülő és nyilvánosság elé kerülő magyar nyelvű szövegeknek illik tisztelniük a helyesírási szabályok szerinti ékezeteket és a sorvégi elválasztást. Ezt talán azok a programozók sem vitatják el, akik egyébként forráskódjaik kommentjeiben változatlanul angolos ékezettelenséggel írják az ékezetes betűket, vagy jó esetben használják az ő, Ő, ű, Ű karakter-kvartett nélküli nemzetközi kódkiosztást, bár minden eszközük meglenne a korrekt magyaros írásmódra is. (Bezzeg nekik könnyű megszokni az elektronikus levelek beírásakor az ékezetek fűnyíró technológiával való leborotválását és az így keletkezett kopasz szövegek visszafejtését, mert előzőleg már jól begyakorolták.)

Abban is szinte mindenki egyetért (mármint abból a szakmai rétegből, amelyet ez egyáltalán érdekel), hogy a programnyelveket érintetlenül kell hagyni, nem szabad azok integritását megsérteni. Itt azonban elég egyetlen lépést megtenni, és máris átérünk egy vitatott területre, hogy milyen nyelven kommunikáljon az operációs rendszer a felhasználóval, milyen nyelvű hibaüzeneteket küldjön.

Igen élesen eltérnek viszont a vélemények a kifejezetten felhasználói programok kezelői felületének nyelvezete körül, és mindkét tábornak vannak figyelemre méltó érvei. E havi összeállításunk írásai is túlnyomórészt ezzel a kérdéskörrel foglalkoznak. Szinte vitaindítónak is tekinthető Kis János írása, aki erőteljesen síkra száll a nyelvileg világszerte egységes kommunikációs közegért, elmarasztalva a számítástechnikai inkompatibilitást és az új Bábel megteremtését. A többi írás zöme viszont inkább arra keres megoldást, hogy a programok a magyarokhoz magyar nyelven szóljanak.

Az előző témáknak mintegy szükségszerű vetülete, hogy akkor milyen is legyen a számítástechnikai szóhasználat. Olyan-e, mint egy orvosi zárójelentésé (amelyből a köznép egyetlen szót sem ért, mégis megtudja belőle, hogy miről van szó) vagy olyan, mint egy politikusi szónoklaté (amelyből a köznép minden szót ért ugyan, mégsem tudja meg belőle, hogy miről van szó).

Sokkal bonyolultabb kérdések ezek, semhogy határozott konklúzióval tudnánk pontot tenni a téma végére. Éppen ezért szívesen vesszük — és közreadjuk —, ha valaki továbbfűzi a felvetett gondolatokat, olyan érvekkel vagy szempontokkal állva elő, amilyeneket szerzőink akár figyelmen kívül hagytak, akár röviden, de ellenkezésre, illetve megerősítésre ingerlően érintettek.



Honosítás, lokalizálás

Út a balek felhasználóhoz

Tanúja voltam a múltkoriban, hogy az Abbruch kifejezést a diákok egyszerűen abruhálásnak „magyarították" az egyik iskolában, ahol német programokat is használtak. Vagy hogy másutt kancellár lett a cancelből. Dehát ettől a játékosságtól kellene annyira megijedni?

Magyarítunk szakkifejezéseket, szakkönyveket, szoftvereket. Bizonyos csoportoknak, szakmáknak kialakult a maguk speciális vegyes nyelvezete. A katolikus egyházét átszövi a latin, a pravoszlávét az ószláv. A nyomdászok szakmai tolvajnyelve német szavakkal van tele, az orvosok a latinra építkeznek, míg a vegyészek és az elektronikai szakemberek az angolból veszik át szakkifejezéseiket. És ez utóbbi lett a sajátja a számítógépes környezetnek is.

Addig ebből nem is származott gond, amíg a gépeket olyanok kezelték, akik magához a szakmához is konyítottak valamit. A számítógép időközben azonban a beavatottak játékszeréből „közfogyasztási cikk" lett. Olyan áru, amelynél az eladhatóság szempontjai előtérbe kerülnek. És mert a világ soknyelvű, és mert a számítástechnikának fontos része a szöveg, az eladást azzal lehet igazán növelni, ha mindenkihez a saját anyanyelvén tudunk szólni, kialakítva a már-már egységes nyelvre berendezkedő számítástechnika új Bábelét.

A kódtáblákkal kezdődött

A kódtáblák lokális háborújával indult az egész, s fokozatosan kialakult a „lokalizált" nyelvi verziók összevisszasága. Sikerült is jól szétcincálni a korábban egységes számítástechnikai rendszereket. Már a billentyűzettel elkezdődik a kavarodás, és tart egészen a programok hibaüzenetéig. És azért közben is van egy két menüpont, help, miegymás.

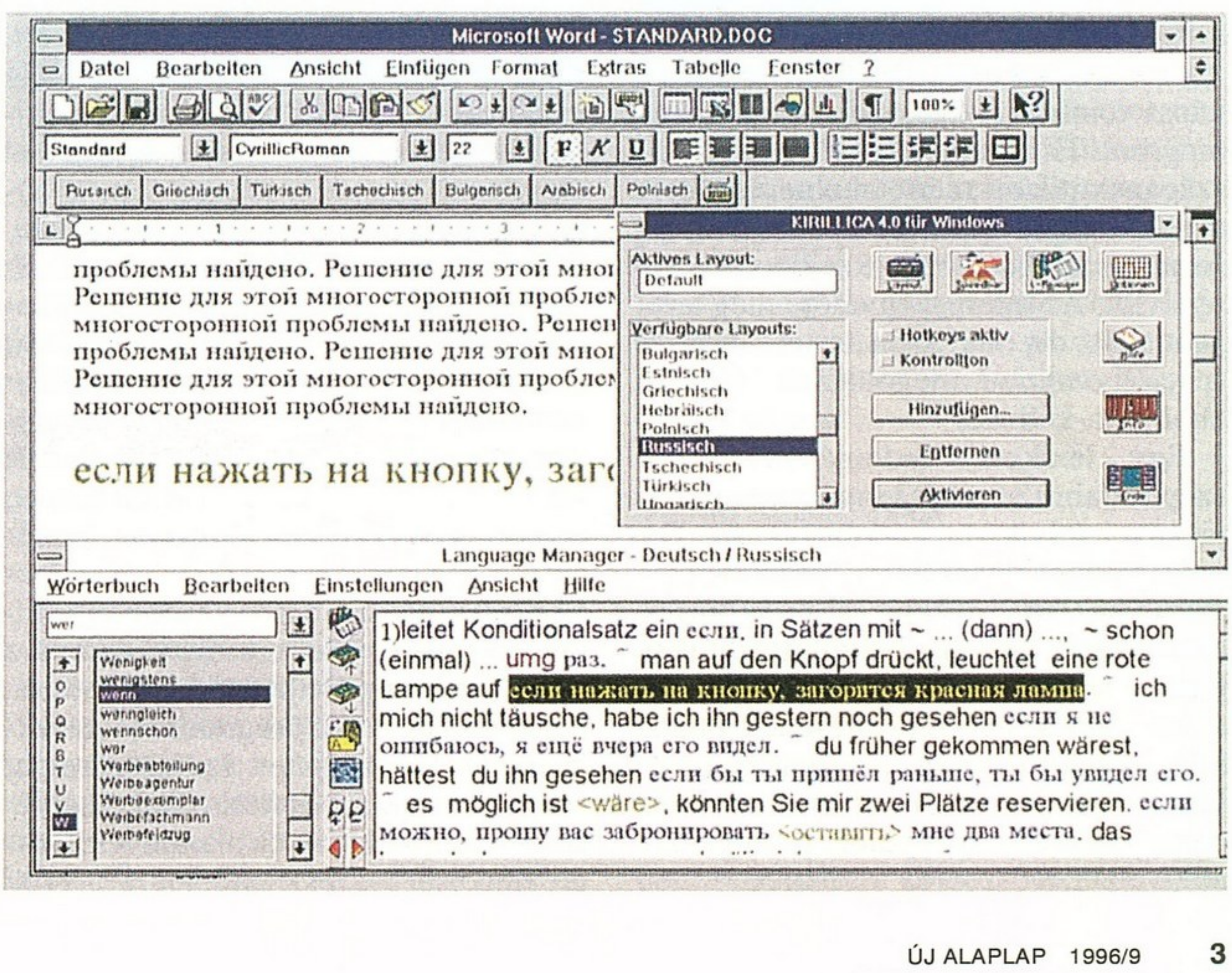
A dolog akkor kezdett komollyá válni, amikor egyes cégek rászánták magukat a túlnyomórészt angol nyelvre épülő programrendszerek lokalizálására (honosítására). Ez tovább bonyolódott, amikor a felhasználók számára készült alkalmazói szoftverek programnyelvi környezetét is lokalizálni kezdték. Ennek azután meg is van az eredménye: például már a WinWord magyar és angol változatának makrói és keretprogramjai sem értik egymást. A parancsszavak lefordítása pedig olyan forgalomnövelő zagyvaságot teremtett, amilyet a kereskedők csak álmukban mertek remélni. Immár külön szakmává vált a kompatibilis programok és makrók írása. Persze az sem rossz, ha valaki fordítóprogramokat tud írni, hogy az egyik nyelvi változat makróit, programjait érthetővé tegye a másik nyelvi változat számára, mert ezzel legalább átmenetileg kialakítható a szoftverek kompatibilitása.

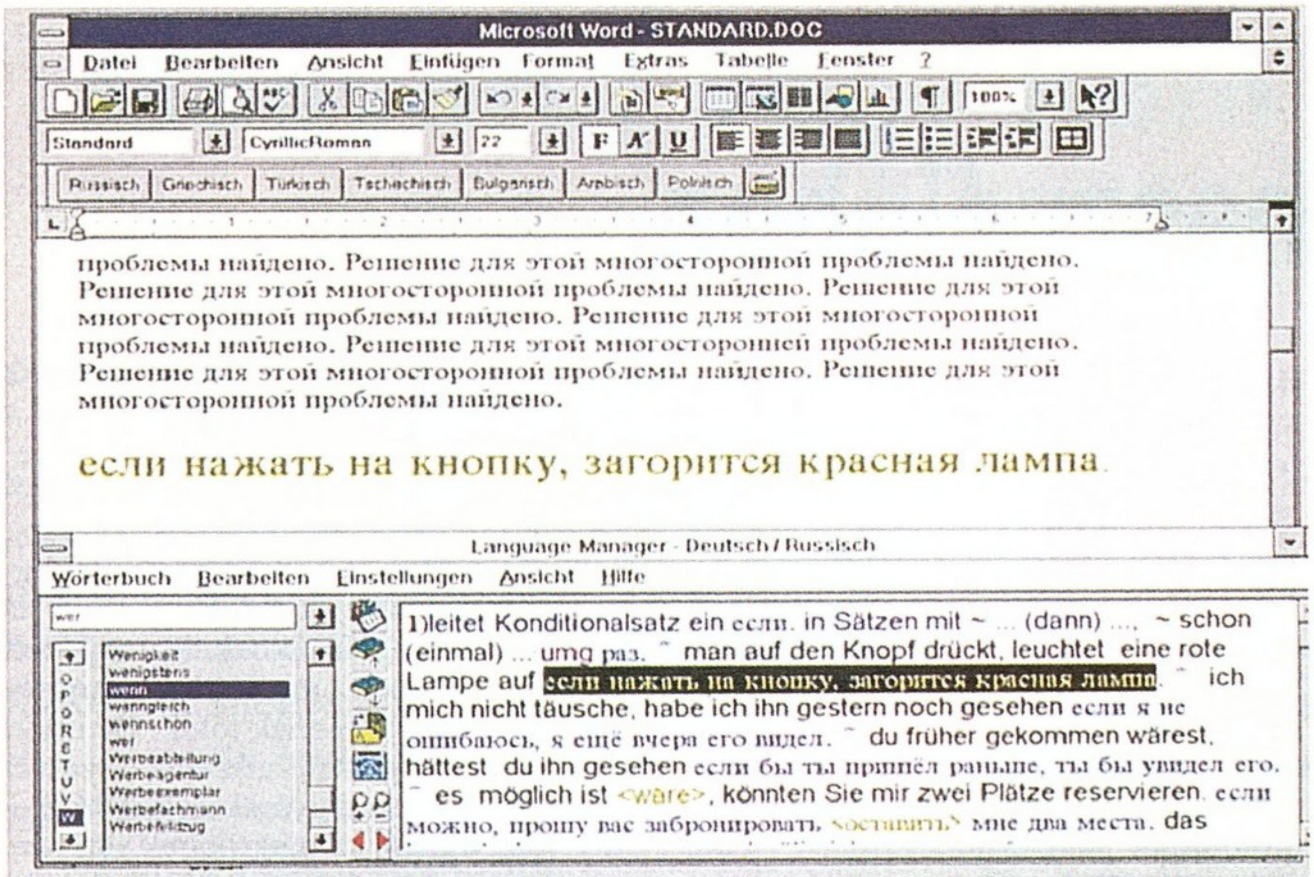
Nyelvvédelem...

A szoftverek nyelvének megújítói gyakran érvelnek azzal, hogy szükség van a nyelvi verziók elkészítésére, hiszen ezzel (piacuk mellett) a nyelvet is védik: megtisztítják az idegen szavak behatolásától. Odáig rendben is van a dolog, hogy a szövegfeldolgozó programoknak kötelességük minden nyelven ékezethelyesen írni és olvasni, szükség esetén ismerni a nyelv jellegzetes karaktereit. A szedő, tördelő és elválasztó programoknak is meg kell tenniük mindazt, amit az anyanyelv törvényei megkívánnak tőlük. De miért kellett a programrendszerek nyelvét bántani? Hiszen azoknak is önálló életük, nyelvtanuk, szintaktikájuk stb. van, ráadásul szigorúbb feltételek között működnek, mint a beszélt nyelvek.

Annak idején a nyelvújítók, később a nyelvvédők időnként keresztes hadjáratot indítottak az idegen szavak ellen. A nyelvújítás érdemeit senki sem akarja elvitatni. De mennyivel lenne jobb, ha minden ötletük megvalósult volna, és az oxigént élenynek, a jódot iblanynak, a nátriumot szikenynek, az alumíniumot timanynak hívnánk? Vagy szerácsnak a patikust, szurdancsnak a tőrt, természetedménynek a fizikát, látodalomnak a színházat, zengeménynek az operát, hangbetűzetnek a kottát?

Az emberi kultúra történetében a számítástechnika különleges nyelvi lehetőséget kapott. Egy szűkebb szakmai réteg legjobb tudása szerint megteremt-





hette, definiálhatta nyelvek teljes szókészletét és nyelvtanát. Minden programozó tudja, hogy a Pascal, a Fortran, a Cobol, a C mind-mind ilyen deklarált szókincsre és nyelvtanra épült.

...és akik védik

Ám ekkor előjöve a Gyártó, és mondá: számomra az a jó, ha eltérek ezektől a világos deklarációktól. És lőn. Az Úr megzavarta Bábel fiainak nyelvét, hogy ne értse egyik a másikát. Az egyikféle Pascal szabályainak megfelelően megírt program nem biztos, hogy lefordítható egy másik fordítóprogrammal. Pedig mindkettő Pascal. Állítólag. Vagy egy magyar AutoCAD Lisp kiegészítése az esetek nagy részében nem megy, mondjuk, a németen vagy az angolon.

A káosz létrejött. A káosz terjed. A káosz rombol. A káosz zsarol. A káosz megront. És a számítástechnika hátországának túlzott mértékű elnemzetiesítése a világot éppen attól akarja megfosztani, amiben az ereje rejlik: a közös nyelvtől. Az Internet egyelőre még tartja magát, de már azon is terjednek a speciális nyelvi megoldások. Ott is, ahol nem kellene.

Egy lexikont, tudományos vagy szépirodalmi munkát természetesen saját anyanyelvünkön legjobb olvasni. De miért kellene ehhez a gép belső nyelvét megváltoztatni? Előbb a nem angol anyanyelvű, de nagy számítástechnikai piaccal rendelkező országok indultak el a programok nemzetiesítésének útján. Mi, magyarok pedig követtük őket. A franciák, a németek, az olaszok kezdték el, majd jöttek a spanyolok, portugálok, lengyelek, oroszok és a többiek.

Való igaz, hogy a nyelvi verziók elterjedésével a számítástechnika populárisabbá vált, mindenki hozzáférhetett. De nem fizettünk ezért túlságosan nagy árat? Az ember és a gép egyre inkább eltávolodik egymástól. Egyre többen tudnak a számítógépről egyre kevesebbet, maholnap immár semmit. Megelégszenek annyival, hogy a gép bekapcsolása után bejelentkezik általában a kedvenc Windows felület. És ha a legapróbb baj van, rögtön szakember és vevőszolgálat után kiáltanak. Kitermelődött minden forgalmazó álma: az applikációs gondolkodású, csőlátásos balek felhasználó.

Divide et impera

Régebben aki számítógépet használt, ritkán vette kézbe a kézikönyvet. (Ha egyáltalán volt neki.) A programok működésének belső logikáját igyekezett megérteni, magán a programon, az alkalmazáson belül kereste a megoldást. És természetesen barkácsolt, kiegészített, ha kellett. Munkaeszközéről, a gépről igyekezett mindent megtudni, mint a régi jó esztergályos szakik. Gyakran volt önmagára utalva, ezáltal kinevelődött egy olyan réteg, amely képes volt alkotó módon hsználni a számítógépet, és azon keresztül megvalósítani önmagát.

A gondolkodó ember azonban mindig veszélyes elem a nem számára teremtett környezetben. Még képes rá, hogy felfedezze a programrendszerekbe tudatosan beépített korlátozásokat, csapdákat, a piacbővítési céllal elhelyezett inkompatibilitások aknáit. Az ilyesmi alaposan összekuszálhatja a nagyok számításait. Tenni kellene valamit „az ügy" érdekében.

A nyelvi verziók kutricái erre nagyon alkalmasnak bizonyultak. Ma már minden sikerorientált cég kénytelen beterelni fejlesztőit ezekbe a szűk ketrecekbe, ha a piacon akar maradni. A felhasználók is meg vannak dolgozva, egyre inkább leszoknak a gondolkodásról, bedőlnek a reklámdömpingnek, és azt is elhiszik, hogy a kézikönyvekben minden le van írva. (Ha valami mégsem működne, tessék megvenni a következő verziót is...) A cégek pedig a sikeren felbuzdulva egyre merészebben manipulálják a jelen és a jövő felhasználóinak agyát. Végül már azt sem lehet tudni, hogy egy szoftver magyar változata melyik angol verziónak felel meg, vagy hogy mi az eltérés közte és az orosz verzió között. Kit érdekel, hogy megy-e az USA-ban vásárolt matematikai program a magyar keretprogramon? Majd elkészül a magyar nyelvű verzió is. Feltéve, hogy lesz rá elegendő igény... Ebben az összefüggésben a nyelv védelmére való hivatkozás közönséges ürügy.

Herz-féle szalámi

Előfordul, hogy több nyelvtudás kell a szoftverek magyarításainak megfejtéséhez, mint az eredeti nyelvű program használatához. Sok esetben még akkor sem derül ki, hogy mit akart mondani a program fordítója, amikor visszafordítjuk a „magyar" szöveget angolra. A németből magyarított szoftverekkel sem jobb a helyzet. A germán tengerikígyó szavakat is könnyebb megérteni az eredetiből, mint a fordításból. Még jó, hogy francia programokkal viszonylag ritkábban találkozunk, mert ők azután úgy megvédik a nyelvüket, hogy szakszótárakban sem lehet nyomára lelni leleményes franciásításaiknak. Sikerült is elrettenteniük minden idegent (és szakmai vitáikból ítélve honfitársaikat is) saját culture de l'ordinateaur-üktől.

Magyarítunk, franciásítunk, oroszosítunk... Még nem tűnt el teljesen a számítástechnikai rendszereket egymással összekötő egységes nyelvi háttér. Egyelőre a programok még nem cenzúrahivatalok és fordítóirodák révén kommunikálnak egymással. Az pedig talán be sem következik, hogy a számítástechnikában épüljenek „berlini falak", mert a folyamat valahol kénytelen lesz megállni. A hexakódban van ugyan két ékezetesíthető karakter, de a nullát és az egyest egyelőre nehéz elképzelni „lokalizált, honosított" változatban.

Kis János

Audiatur et altera pars

Legújabbkori nyelvújítás

A 90-es évek elején űrkori honfoglalás vette kezdetét: a kibertérben, a számítógépekkel és számítógépes programokkal benépesített mesterséges világban kell helyet keresni magunknak. Ebben az új „honfoglalásban" a magyar nyelv azért is kulcsszereplő, mert a számítástechnikán keresztül részben a köznyelv is elangolosodhat. Mit tehetünk általában, és mit tesznek konkrétan a Microsoft szoftverei e tendencia ellen?

Egyes országok törvényileg korlátozzák az idegen nyelvű szoftverek forgalmazását: Franciaországban erről törvény is született. Ilyen jellegű „önkéntes COCOM-törvény" Magyarországon természetesen szóba sem jöhet. Az egyetlen lehetőség a számítástechnikai szakemberek összefogása: együtt küzdeni a „kopi-pésztek", „brózerek", „ímélek" ellen.

Elittudomány, vagy...

Sajnos a jelenlegi hazai számítástechnikai elit általában elutasítóan viselkedik a magyar (vagy magyar nyelvű) szoftverekkel szemben. Ha megkérdezünk egy rendszergazdát, szakértőt, szoftverkritikust, mi a véleménye a magyar szoftverek, és ezzel együtt a magyar terminológia használatáról, kevés pozitív választ kapunk.

Az ellenérvek egy része kétségtelenül megalapozott és jogos: sok fontos program angol nyelvű, és azok zavartalan működése csak „vegytiszta" angol környezetben biztosítható. Ugyanakkor a sablonos kategórikus elutasítást közönséges számítástechnikai sznobizmusnak tartom, főleg az olyan megnyilvánulásokat, hogy: „azért nem használok magyar programokat, mert azok mindig rosszabbak", „soha semmit nem találok meg rajtuk", „egyszerűen nem tudom megszokni", „nevethetnékem támad, ha magyar menüpontokat látok" stb.

Pedig ha nem szervert futtatunk, nem fejleszteni akarunk, nem célszoftvereket használunk, hanem például csak levelet írunk és küldünk, táblázatot kezelünk, fájlokkal teszünk-veszünk, Interneten böngészünk — vagyis a számítógépek alkalmazásának legtöbb területén —, bármiféle funkcióveszteség nélkül használhatunk magyar programokat. És a behatárolhatatlan „átlagfelhasználó" (elnézést a kategorizálásért) szerencsére használ is, míg a jól behatárolható véleményalkotó és véleményformáló számítástechnikai elit kollektív elutasítást tanúsít.

A fenti, végül is szubjektív tényezőkre visszavezethető visszatartó erő mellett azonban van egy nagyon is objektív korlát, maga a honosítás és honosíthatóság problémaköre. A számítástechnika kezdetekor azért nem létezett ez a probléma, mert minden angolul volt, és aki ezt a tudományágat akarta művelni, annak bizonyos fokig ugyanúgy meg kellett tanulnia angolul, ahogy egy orvosnak latinul. Csakhogy a számítástechnika ma már nem elittudomány, hanem üzlet, a mindennapok kelléke. Egy tömegterméknél pedig kifejezetten hátrány, ha az angolul nem értőket távoltartja magától.

A 90-es évek elejétől minden nagyobb szoftvergyártó felismerte a honosítás szükségességét, és napjainkban elértünk oda, hogy a soknyelvűség igényét már a program tervezésekor is figyelembe veszik. A sikeres honosítás létrejöttének ez azonban csak szükséges, de nem elégséges feltétele, és ez különösen igaz a magyar nyelv esetében!

A honosítás problémakörét három fő csoportra bonthatjuk: a honosíthatóság, a szókincs megteremtése és a tényleges fordítás problematikájára. A szoftver honosíthatósága — ami különösen a rendszerszoftvereknél alapkövetelmény — nem is olyan magától értetődő dolog. Tulajdonképpen arról van szó, hogy programunk alapvető szinten képes legyen a különböző nyelvi jelek megfelelő kezelésére, minden információt — nyelvtől függetlenül — be tudjunk táplálni, meg tudjunk jeleníteni és fel tudjunk dolgozni a számítógépen. Emellett fontos követelmény, hogy a rendszer lehetőleg támogassa a különböző nyelvi szabványokat, normákat.

Univerzális megoldás?

A fenti követelmények kielégítése jegyében születtek a különböző nyelvek kezelésére szolgáló kódlapok, karaktertáblák. Eddig a legelterjedtebb megközelítés az volt, hogy a rendszer 8 biten tárolja a karaktereket. Ez 256 karakternek elegendő hely, így egy ügyesen választott karakterkészlettel akár több nyelvet is lefedhetünk. Az viszont, hogy melyik 256 karakter hol helyezkedjen el, már nem egységes a különböző szabványok között: Magyarországon a hivatalos szabvány az ISO 8859-2 karaktertábla, a de facto szabvány viszont a Windows kódlap. (Szerencsére a magyar-specifikus karakterek ugyanoda kerültek a két készletben.)

A 8 bites karakterek azonban nem hozhatnak hosszú távú megoldást. Ezzel a ugyanis egy új problémát is létrehoztunk, nevezetesen: hogyan tudnak együttműködni az eltérő kódlapok. Az ideális megoldás egy egységes karakterkészlet használata lenne. A szabvány — a Unicode karakterkészlet (ISO/IEC 10646-1) — már régen meg is született, sőt a gyakorlatban is megjelent például a Microsoft Windows NT-ben.

Bár nem elvi okok miatt, de sokszor az alapvető karakterkezelési problémákkal megegyező súllyal eshet latba a nemzeti szabványok hiányos vagy nem megfelelő kezelése. Példaként említhetjük a dátumformátum különbözőségét. Ez sajnos minket is sújt: számos program fel sem tételezi, hogy a dátum esetleg nem az angol formátum szerint jelenik meg. Ez még működésbeli eltérésekre is vezethet, ha például pont szerepel a dátumformátumban vagy a dátum végén.

Magyar szakszókincs

A szókincs problémája nagyon egyszerűen megfogalmazható, de nehezen megoldható. Bár a számítástechnika születésekor és fejlődésekor végig jelen voltak a magyar tudósok is, a számítástechnikának csak új keletű nyelvi hagyományai vannak Magyarországon. Az angol nyelv dominanciájával szembenézve olykor merész nyelvi lele-

ménnyel, máskor inkább konzervatív módon, de folyamatosan meg kell alkotnunk a számítástechnika magyar terminológiáját.

Honnan meríthetünk? Elsődleges forrásként a magyar programírók fejlesztései szolgálhatnak. Sajnos ezek csak kis részben fedik le azokat a területeket, amelyek megfelelnek napjaink uralkodó számítástechnikai irányzatainak. Leggazdagabb forrásunk lehetne a számítógépet használók nyelvi alkotóképessége.

Ez az erő időnként valóban nagy segítségünkre is van, de sajnálatos módon inkább csak ott bukkannak fel a javaslatok, ahol egyébként is könnyű magyar megfelelőt találni. Senki sem ütközik meg az egér, a merevlemez, a hangkártya szavak hallatán. Ugyanakkor tanácstalanul állunk olyan kifejezések előtt, mint az online, az e-mail, vagy a plug and play...

Újabb kihívások

A Microsoft Magyarországnál három nagy hullámban jelentkeztek a nyelvi honosítási törekvések. A legelső fontos lépés a Windows 3.1 magyar fordítása volt, úttörő jelleggel a magyar alapterminológia megalkotásában, a Microsoft elismertségének megalapozásában és a honosítási problémák felismerésében. Annak idején heves csatározások folytak, mi legyen a file-lal, a helppel és egyéb sarkalatos szavakkal. E két szó nagyon jól példázza a rendelkezésünkre álló játékteret: a Help laza asszociáció révén kitűnően helyettesíthető a Súgóval, míg a file esetén sajnos be kellett fogadnunk egy idegen testet, így született a fájl. Az egyetlen vetélytárs az állomány szó volt, amelyet végül mégis elvetettünk, mert a beszédben a fájl már túlságosan elterjedt.

A szóalkotás második fázisában, 1994/95-ben a Windows 95 lehetőséget nyújtott arra, hogy finomítsuk a korábbi terminológiát, tisztázzuk a pontatlanságokat. Emellett sok új elem is megjelent a programban, ami újabb kihívást jelentett. A két legnehezebb falatnak az e-mail és az online bizonyult. Az előbbire volt javaslat: elektronikus posta vagy elektronikus levél. A probléma ezzel csak az, hogy használatát tekintve egyik sem az e-mail, hanem az electronic mail megfelelője. A rövid változatra máig sincs elfogadható magyar javaslat. A jelöltek közül (emíl, drótposta, e-posta, e-levél, fényposta, villanylevél) egyik sem elég megnyerő. Az online esetében viszont még vitaalapnak alkalmas javaslat sem volt.

A Windows 95 honosításának egyik kulcsszava a folder. Két konkurens szó közül végül a józan megoldás, a mappa került ki győztesen. A másik változatot, a tasakot, bár kitűnően leírta volna a folder funkcióit, végül érzelmi töltete miatt vetettük el. A mappa mellett szólt még az az érv is, hogy ez a fordítás már korábban is megjelent, nevezetesen a Macintosh magyar változatában.

Hol a határ?

Szóalkotási törekvéseink harmadik állomásán a Microsoft Magyarország 1996-ban jutott túl. Meg kellett alkotnunk az egyre köznapibb Internet, illetve a továbbra is erősen szakmai jellegű Windows NT terminológiáját. Az Internettel kapcsolatos szavak esetében segítségünkre szolgált a magyar Internet-társadalom, amely már évek óta ízleli, próbálgatja az új magyar szóalkotásokat (például: honlap, ottlap, otthonlap stb.), illetve az átferdült szavakat (amilyen például a hipertext).

A Windows NT esetében a felhasználás technikai jellege fogós feladatot adott. Nem biztos ugyanis, hogy érdemes belevágni olyan erőszakos fordításba, amelynek eredményeként a magyar terminológia megértési problémákat okozhat, hiszen olyan régiókban, ahol csak tapasztalt felhasználók járnak, esetleg többet árthat egy rossz fordítás, mint a nem fordítás. Két példa. A „domain" szó központi szerepet játszik az NT hálózati működésében, természetes tehát, hogy arra magyar szót kell használnunk (tartomány). Ugyanakkor a „named pipes" kifejezés (a programozók tudják, miről van szó) angolul fog szerepelni, mert ennek lefordításával nem nyernénk semmit, viszont elérnénk azt, hogy a hozzáértők sem tudnák, miről van szó.

Nyelvszerkezeti buktatók

A honosítás sokszor igen nehezen áthidalható problémákat tartogat a fordítók számára. Az angol nyelv egyszerű szerkezete, rövid szó- és mondathosszúsága, szavainak sokféle jelentése, sok szó igei és névszói jellege, egyszerű szabályokra épülő nyelvtani kapcsolódása kitűnő talajt jelent egy szoftver felhasználói felületének megírásához, hiszen itt elsődleges szempont a tömörség és a többcélúság. Sajnos (vagy olykor inkább szerencsére?) e jegyek nem találhatók meg a magyar nyelvben, ezért azokat magyarra igen nagy leleménnyel vagy adhoc megoldással lehet csak átültetni.

Mik a legfogósabb szerkezeti különbségek? A magyar agglutináló nyelv, ami azt jelenti, hogy túlnyomórészt toldalékok segítségével fűzi mondatba a szavakat, nem pedig elöljárókkal, ahogy az angol vagy a többi indoeurópai nyelv teszi. Ennek kapcsán tipikus probléma adódik például a dátumkifejezések írásánál. Az angolban ezt találjuk: „every 2nd week on Tuesday". Vagyis: „minden második hét keddjén". A probléma ott kezdődik, hogy a többcélúság miatt a hét napjai általában az operációs rendszerből származnak, mi viszont nem tudunk egységes ragot hozzáilleszteni a különböző napokhoz. A megoldás egy körülményesebb, de általánosan mégis alkalmazható fordítás: „minden második hét keddi napján".

A másik klasszikus feladat az angol szenvedő szerkezetek fordítása. A magyarban teljesen háttérbe szorultak a nyelvtani szenvedő szerkezetek, a szoftverekben viszont az angol nyelv az átlagnál gyakrabban használja.

Legyünk elnézőbbek

További nehézségek elé állítja a fordítót az olyan szavak fordítása, amelyek igeként és névszóként is szerepelhetnek, vagy többértelműek. Az angolban megengedhető az a nagyvonalúság, hogy ugyanazt a szót használják mindkét esetben, mi viszont... vadászhatunk az ilyen jellegű hibákra a készülő termékben.

Bár saját gyengéinket inkább kendőzni illdomos, mégis idézek egy példát a tanulság kedvéért. A Windows 95-ben van egy párbeszédpanel, amely akkor jelenik meg, ha egy helyigényes művelet végzése közben betelik a merevlemez, de a Lomtár még nem üres, tehát van felszabadítható terület. Ekkor a program megkérdezi, akarom-e véglegesen törölni a Lomtár tartalmát. Az erre parancsot adó gomb neve helyesen „Ürít" lenne, nem pedig „Üres". Angolul ugyanis „Empty" a gomb neve...

Félreértés ne essék, a fenti problémák nem magyarázkodásként vagy indoklásként szolgálnak. Meggyőződésem, hogy a legtöbb alkalmazási szituációban a magyar termékek ma már kitűnően használhatók, ugyanis a feladat nehezén — a honosíthatóság legfőbb gátjain — már túljutottunk. A honosított szoftverek választása a legtöbb esetben sok előnnyel jár. Ha pedig mégsem teljesen szabatos nyelvezeten szólalna meg a program, legyünk vele szemben elnézőbbek. A honosítás ügye érdekében...

Barkóczi Miklós

A Nagy Kéknél is változnak az idők...

Honosítás vagy fordítás?

A magyar régebben nem szerepelt az IBM által célnyelvként megjelölt listán. Az OS/2 Warp magyarított változatának megjelenésével megtört a jég: ma már az IBM számára sem közömbös, hogy mi zajlik a magyar piacon. Érdekes ezzel kapcsolatban megnézni, mi történt az utóbbi években az IBM házatáján a honosítással kapcsolatban.

Öt éve az IBM hatalmas pénzeket fizetett ki belső fordítógárdájának fenntartására, nem utolsósorban az idegen nyelvű változatok előállításával kapcsolatos fordítási tevékenységre. Ekkoriban hozzávetőlegesen évi egymillió oldalt fordítottak, ami a cégnek kb. 150 millió dollárjába került. 150 dollár oldalanként! Hiteles források szerint egyetlen svéd szó előállítása 2,60 dollár költséget jelentett az IBM-nek. (Más cégek ennél jóval kevesebbet fizettek a fordításért: alig egyharmadát oldalanként. De volt olyan cég is, ahol nyolc oldalt fordítottak le annyiért, amennyibe az IBM-nek egyetlen oldal fordítása került.)

Változó „külpolitika"

Ez a nagyvonalúság is hozzájárult ahhoz, hogy rövidesen bekövetkezett a krach: az IBM az amerikai üzleti élet egyik legnagyobb veszteségét könyvelhette el. Nyomoztak, hogy mire mennek el a százmilliók, és rövidesen nekiláttak gyökeresen átalakítani a cég egész fejlesztési és üzletpolitikáját. A változás a honosítási politikát és a fordítások készítését sem kerülhette el.

A legnagyobb tételt itt a nagy nyugat-európai nyelvek jelentették — ezek fordítási munkáit sürgősen áttelepítették a „célországokba". Továbbá felállították Koppenhágában egy központot a nyelvi problémák koordinálására és összehangolására, az Egyesült Államokban pedig létrehoztak egy külön szervezetet a nemzeti nyelven folyó fejlesztések támogatására (National Language Service).

Az új koncepció az volt, hogy a helyi központok közvetlenül kapják meg a fordításhoz szükséges anyagot a fejlesztő laboratóriumokból, de problémáik megoldásáról — mindenütt a helyi adottságoknak megfelelően — maguk gondoskodjanak. Ahol lehet, kössenek szerződéseket külső szervekkel, ha kell, szervezzék meg saját fordítógárdájukat. Az IBM központilag ellátja instrukciókkal a helyi szerveket, de gazdaságilag meghagyja önállóságukat.

A fordítások gépi segítése

Az új rendszer kialakítása nem ment zökkenők nélkül, de egészében véve bevált. Hamarosan kiderült viszont, hogy sokkal nagyobb szükség van a fordításoknál a gépi segítségre, mint azt előzőleg gondolták. Igaz, hogy az általánosan használható gépi fordító rendszerek ideje még nem jött el, ahol azonban hasonló jellegűek, sőt hasonló megszövegezésűek a szövegek, és ezekből kell nagy adagokat lehetőleg gyorsan lefordítani, ott még egy korlátozott tudású fordítástámogató rendszer is óriási segítséget jelenthet.

A számítástechnikában tipikusan ilyen szövegekről van szó, hiszen például egy-egy új programváltozat megjelenésekor a dokumentáció tekintélyes része egyáltalán nem változik. A korrekciókat viszont nagyon következetesen kell végigvezetni minden érintett szövegrészen, és például a terminológia egységességének, az esetleges változtatások következetes érvényesítésének biztosítására speciális automatizmusok kidolgozására is szükség lehet.

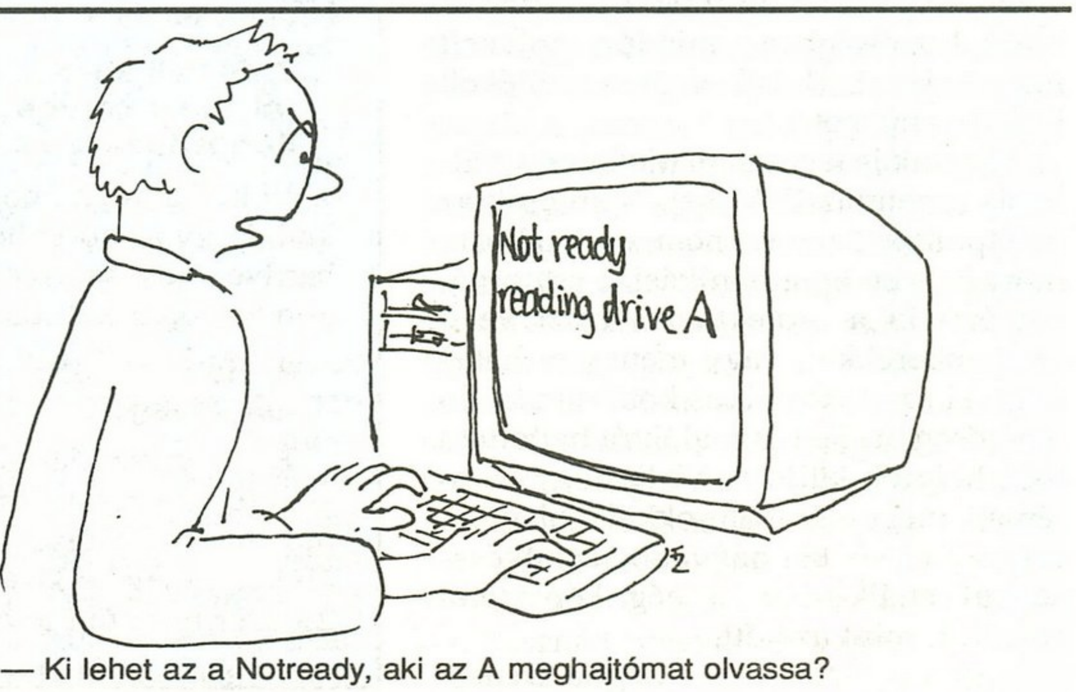
Eleinte saját problémáik megoldására fogtak bele egy olyan eszköz kifejlesztésébe, amelyből később azután piaci termék is lett. A világon ma már sok helyen használják (például a különböző nyelvekre honosított, lokalizált szoftverek előállításában) a „Translation Manager"-t. Ez a fordításmenedzselő rendszer számos olyan funkcióra képes, amely fordítás közben szükséges lehet:

— Bele van építve egy mindentudó, speciálisan a fordítói igények kielégítésére készített szövegszerkesztő. Ez párhuzamosan mozgatja a forrásnyelv és a tárgynyelv meglévő szövegrészeit, és előre-hátra meg tudja keresni a hasonló szövegkörnyezeteket. (Így például az új terminológia kialakítását segítheti azzal, hogy előre meg lehet nézni, másféle szövegösszefüggésben is előfordul-e valamelyik újonnan felbukkanó kifejezés.)

— Kezeli a szótárakat, tematikus szógyűjteményeket és terminológiai kifejezéstárakat, beleértve a keresésen és olvasáson kívül ezek karbantartását, javítását, bővítését is (természetesen jelszóval védve őket a jogtalan hozzáféréstől).

— Helyi hálózaton keresztül biztosítja az erőforrások megosztását és az együttes munka összehangolását, miközben a fordító a helyi eszközök használatán kívül, velük egyidejűleg hálózatba kapcsolt távoli szótárakhoz is hozzá tud férni.

— Talán a legfontosabb komponens egy olyan háttérben működő szöveges adatbázis, amely párhuzamos, „előre-



— Ki lehet az a Notready, aki az A meghajtómat olvassa?

gyártott" nyelvi paneleket tárol egymásnak megfelelő szövegrészekből, mondatokból és mondatnál kisebb szócsoportokból. A szakmában fordítástárként, „Translation Memory"-ként szokták emlegetni ezt az eszközt, amelyet azóta más hasonló programtermékek is széles körben használnak.

A Translation Memory talán a legfontosabb újdonsága a rendszernek. Az alapötlet tulajdonképpen nem új: tudomásom szerint Don Walkertől származik, aki már a 80-as években kidolgozta egy olyan fordítástámogató rendszer tervét, amely jó fordítások mondataiból, ezeknek összefüggő szeleteiből nagy szöveges adatbázisokat tud felépíteni. Az így kapott nyelvi panelek különböző kulcsokkal bármelyik nyelvre vonatkozóan adatbázisból lehívhatóvá tehetők, sőt bizonyos hasonlósági kritériumok teljesülése esetén automatikusan is megjelenhetnek a fordító képenyőjén.

Szervezési kérdések

Az IBM Translation Managere jelenleg 19 nyelv fordítását támogatja a némettől a kínaiig és japánig, az izlanditól a katalánig, oda-vissza. Egyirányú felhasználási céllal már a magyarra, a csehre és a lengyelre való fordítást is ajánlgatják. 15 floppyn vagy CD-n elérhető az angol, francia, német és olasz változat. Fut Warpon, 3.1-es és 3.11-es Windowson és Win95-ön, elfogad tízféle dokumentumtípust a PageMakertől a Quarkig, a WinWordtól a HTML-ig.

A honosítás problémája szorosan összefügg a fordítások kérdésével, nem azonos vele. A honosítások tekintélyes részét ma is házon belül oldja meg a cég, de sokkal olcsóbban és hatékonyabban. Ma már az IBM-nél úgy látják, hogy a szervezést igen, de az elvi kérdések eldöntését nem szabad rábízni egyetlen emberre. Az ilyen témát előzőleg alaposan meg vitatni egy széles körű bizottságban, minden érdekelt részvételével. (Lásd erről a mellékelt keretes anyagot.)

Magunk is tapasztalhatjuk, hogy változás történt az IBM-nél. Nem csak az üzletpolitikában, és nem csak abban, hogy milyen kommunikációs csatornákat épít ki a projektjében részt vevő szakemberekkel, vagy mennyire használja ki az elektronikus kommunikáció lehetőségeit. Felhasználóktól hallottam több helyről (külföldön is!), hogy problémáik megoldásában sokkal több megértéssel és sokkal nagyobb segítőkészséggel találkoznak a cég képviselői részéről, mint azelőtt.

Vargha Dénes

Honosítási szempontok

A honosítással összefüggő problémákat és megoldandó feladatokat Deborah Brock Webb (aki hosszabb ideig volt az IBM ez irányú tevékenységének vezetője) a következőképpen látja és hangsúlyozza:

1. Először is el kell fogadtatni a menedzsmenttel, hogy neki kell viselnie a költségeket.

Régen rossz, ha a menedzsment nem támogatja a nemzeti változatok létrehozását. Ha nemzetközivé akarunk válni, meg kell fizetnünk az árát.

2. Kezdettől fogva éberen kell figyelni a fejlesztés folyamatát.

Ha már angolra sem megy simán a dolog, más nyelveknél hatványozottan jelentkeznek a problémák.

3. A honosításnak kell a figyelem középpontjába kerülnie.

Egy számítástechnikai cégnél megkérdezték az egyik vezetőt, hogyan lett ő a projekt honosítási vezetője. „Szabadságon voltam, és rám osztották ki." Ez komolytalan dolog. Idejében gondoskodni kell róla, hogy igazi gazdája legyen a honosítási projektnek.

4. Megfelelő kommunikációs csatornát kell kiépíteni minden résztvevővel.

Akár házon belül, akár külső szakemberekkel akarjuk megoldani a honosítást, a tervezés legelejétől minden részfolyamat összes résztvevőjét idejében és korrektül be kell vonni a munkálatokba. A cég szoftverfejlesztő mérnökeivel eleve megvan a kapcsolat, velük nincsenek kommunikációs problémák. Ott kezdődik a baj, ha ők kimaradnak a honosítás megtervezéséből, mert nélkülük üressé, formálissá válik a távoli fejlesztőkkel fenntartott kapcsolat.

5. Ki kell alakítani a megfelelő terminológiát.

A termék legapróbb részleteire is ki kell dolgozni egy konzisztens terminológiát — és ez éppúgy érvényes az angolra, mint más nyelvekre. Biztosítani kell, hogy mindenki következetesen tartsa magát az egységes terminológiához minden célnyelvben. Az egyértelmű azonosíthatóság lehetőségét nyelvi szinten is meg kell teremteni, ezért például kerülni kell az amerikai szleng használatát.

6. Nemzetközi fejlesztő eszköztárat kell létrehozni.

Biztosítani kell, hogy minden nyelven és minden kultúrközegben rendelkezésre álljon és használható legyen minden hozzávaló. A fejlesztéshez, fordításhoz és publikáláshoz szükséges technikai és szoftvereszközök teljes fegyvertárán kívül ebbe beleértendők maguk a felhasználandó információk is. Kerülni kell a saját gyártmányú eszközök használatát, mivel ez növeli a betanításra és a működtetésre szánt költségeket.

7. El kell készíteni a termék tesztelési tervét és a tesztelési folyamat pontos leírását. Ebben részletesen ki kell térni a honosításhoz szükséges tesztelés lépéseire is.

El kell dönteni, hogy mely pontokon kerüljön sor a tesztelésre. Biztosítani kell, hogy az adott helyen és időpontban rendelkezésre álljanak a szükséges hardver- és szoftvereszközök. Ha a tesztelést helyben végzik, mindig gondoskodni kell tartalék erőforrásokról is.

8. Ha csak lehet, ki kell használni az elektronikus kommunikáció lehetőségeit.

Ez egyrészt javítja a kommunikációs kapcsolatot a munka résztvevői között, másrészt lehetővé teszi nagy információs tömbök átvitelét is. A modem kapacitását sem szabad figyelmen kívül hagyni. (Kétmilliónyi sor átvitelét meg se kíséreljük olyan modemmel, amely másodpercenként csak 28 ezer bitet tud továbbítani.)

Honosítási tapasztalatok

Nehezebb is, könnyebb is

A szerző hosszabb ideig dolgozott Írországban, a Microsoft honosítással foglalkozó központjában. Munkája mellett sok beszélgetésre is alkalma nyílt a honosítás legkülönbözőbb kérdéseiről számos más országból jött munkatársával. Dublinban szerzett tapasztalatai nagyban hozzájárultak jelenlegi nézeteinek kialakulásához.

Lássuk, hogyan történik egy szoftver honosítása. Azt elég korán megtanuljuk, hogy a szoftver futtatható programkódból és adatokból áll. Az adatok egy része független a felhasználó nyelvétől és kulturális hátterétől (nevezzük ezeket rendszeradatoknak), mások viszont függenek tőle, mint amilyenek például a felhasználói felület építésében vesznek részt: az üzenetek szövege, az ablakok feliratai, valamint a dátumformátum, a használt pénznem stb.

A feladat egyszerű: a programkód és a rendszeradatok érintetlenül hagyásával a felhasználói felületen megjelenő, angol nyelvű adatokat le kell választani a szoftverről, le kell fordítani, majd vissza kell tenni a programba immár magyar nyelven.

Hát akkor kezdjük!

A honosítás többnyire a gyakorlatban is így történik. Ezenkívül persze meg kell vizsgálni a kulturális háttér által befolyásolt adatokat, és ahol szükséges, meg kell őket változtatni. És az új szoftvert természetesen újra kell tesztelni, hiszen teljesen váratlan helyeken ugorhat ki a szoftver bármely részében lévő hiba.

A fentiekből is látszik, hogy a szoftver akkor lesz könnyen honosítható, ha már a fejlesztéskor figyelembe vesznek néhány szempontot az alkotók. Ezek közül a legfontosabb az, hogy le lehessen választani a nyelv- és kultúrkör-specifikus adatokat a szoftverről. A Microsoftnál, részben saját hibáikon is okulva, már évek óta ennek az elvnek a szem előtt tartásával írják a programozók a programokat. Az amerikaiak köztudottan nem híresek nemzetközi gondolkodásmódjukról, azt a szabályt azonban szigorúan betartják, hogy a képernyőn megjelenő szövegeket a könnyű honosíthatóság érdekében külön helyen kell tárolni a szoftverben. Erre egyébként jó eszköz a windowsos környezet: az úgynevezett resource részben lehet szerepeltetni minden ilyen adatot, ez a rész pedig egy .EXE fájlban igen egyszerűen lecserélhető.

Építőkocka felnőtteknek

A következő tervezési szempont, amelyet egy jól honosítható szoftvernek követnie kell, azokra az üzenetekre vonatkozik, amelyeket futásidőben kell a szavakból mint építőkockákból kialakítani. Ilyen üzenetre példa a következő mondat: „The name of the directory has changed to X". Itt a megjelenítendő üzenet az X előtti részből és a könyvtár nevéből alakul ki, közvetlenül a megjelenítés előtt.

Ezek a mondatok azért veszélyesek, mert óhatatlanul beléjük van égetve az angol nyelv néhány szintaktikai szabálya. Ha például az előző mondat úgy készül el, hogy a program a mondat végéhez hozzáilleszti a könyvtár nevét, akkor a fordító nyilván nem választhat olyan fordítást, amelyben a könyvtárnév a mondat közepén van. Könnyebb a helyzet, ha a könyvtárnév egy paraméter, amelyet fordításkor mondjuk egy % jel helyettesít a mondatban. Ilyenkor a fordító bárhová teheti a megfelelő jelet a mondatban.

A ragokat a szótő hangrendjéhez igazító magyar nyelv esetében azonban ez sem elég, hiszen nem tudjuk megadni, hogy a szoftver hogyan válasszon a „-ra" vagy a „-re" rag közül. Itt csak az segítene, ha a szoftver meg tudná határozni a hangrendet, a fordító pedig meg tudná adni a választható ragokat, az alkalmazási szabállyal együtt. Ilyen mértékű függetlenség az angol nyelv szintaktikájától természetesen még csak álom, de derék amerikai barátaink megértették, hogy a fordítóknak lehetőséget kell kapniuk az üzenetek átszervezésére, és ha lehet, ehhez tartják magukat.

Ne gondoljuk, hogy magyarul minden sokkal hosszabb és rosszabb, mint angolul. Az, hogy hosszabb, az esetek többségére valóban igaz. De igenis vannak olyan pontok, ahol találóbb, szebb és rövidebb megoldásokat tesz lehetővé a magyar nyelv, mint az angol. Sokan ezt azért nem veszik észre, mert a magyar szövegeket is az angol nyelv logikájára próbálják felépíteni, ahhoz pedig természetesen az angol igazodik a legjobban.

Áldozati egér

Nemzeti büszkeségünket tovább legyezgetendő, hadd mondjam el, hogy felhasználói terminológiában a magyar sok más nyelvnél sikeresebben váltja ki honi kifejezésekkel az angol szakszavakat. Ne higgyük, hogy a magyar nyelv alkalmazható a legnehezebben a szoftver és ember közötti kommunikációra. Csak példaképpen: nekünk nagy könnyebbség, hogy nem kell bajlódnunk a nemekkel. Képzeljük el, mondjuk, a fenti beillesztős probléma különböző variációit a lengyelben, ahol a nemek egyeztetését is figyelembe kell vennie a fordítónak.

Sokszor nehéz eldönteni, hogy mi a rendszeradat, és mi a felhasználói adat — ezt azonban a programozói figyelmetlenség is okozhatja. Jó programozó például nem használja ugyanazt a stringet ennyire különböző két célra. Egy amerikai programozó barátunk világosan megértette, miért kell erre vigyázni, mégis előfordult vele, hogy egy helyütt eltévesztette a Windows 3.1-ben. Ennek a figyelmetlenségnek a következménye, hogy a Telepítő magyar változatában egyszer angolul jelenik meg a képernyőn a „mouse" szó, mert a rendszer a képernyőn megjelenítendő stringet használja az egyik .INI fájlban egy bejegyzés létrehozására. Ebből az a kalamajka származott, hogy ha megváltoztattuk a szót „egér"-re, akkor az INI fájlban egy „egér=" sor keletkezett, amelyet a Windows többi része nem tudott értelmezni, tehát meg kellett tartanunk az angol szót — ez biztosította a hibátlan működést. Viszont méltán vethetik a szemünkre, hogy miért keveredik angol szó a felhasználói felületen használt magyar szövegbe.

Varga Gábor

Több szem többet lát

Fordítói műhelymunka

Amikor felkértek, hogy az Új Alaplap olvasói számára a szoftverek honosításának gyakorlati problémáiról írjak, örültem, hogy végre lehetőségem lesz kifejteni véleményemet azoknak, akik minden szoftvermagyarítást fenntartással fogadnak. Később, amikor nekikezdtem írni, elbizonytalanodtam, hogy elég érdekes lesz-e a mondandóm. Felhatalmaztam hát a szerkesztőséget, hogy húzza ki az érdektelen részeket. Ha tehát az olvasó mégis talál ilyesmit, én mosom kezeimet...
(Ezt a felelősségátháritást egyébként nem én találtam ki. Minden szoftveres licencszerződés úgy kezdődik, hogy ha felbontotta, elfogadta a feltételeket.)

A legtöbb nyelv — ilyen a magyar is — megpróbálja kivédeni az idegen szavak mértéktelen beáramlását. A tudomány és a technika fejlődése, az újabb tárgyak előállítása vagy a létező világ részletesebb leírása (esetleg a nyelvi divat) időnként azonban új szavakat igényel. A tudomány és a gyakorlati élet egyes területeinek leírására a köznyelvitől szóhasználatában eltérő szaknyelv alakult ki. A szoftver honosításakor jellemzően két ilyen csoportnyelv szókincsére érdemes odafigyelni. Az egyik a számítástechnika általános nyelvhasználata, a másik az adott szakma nyelvezete — például a szoftvernél az ügyvitel és a pénzügy, a szövegszerkesztőknél a kiadvány-előállítás. De össze lehet-e ezeket egyeztetni?

Arra is tekintettel kell lenni, hogy a legtöbb szoftvert széles körű, nem csupán egy szűk szakmai kört érintő használatra szánták (s éppen az ilyenek magyarítására van esély és kereslet), ezért meg kell találni a szakszerűség és a közérthetőség kompromisszumát. Ennél sokkal több gondot szokott okozni az, amikor egyes köznapi szavak jelentését valamilyen irányba szűkíteni szükséges, és az adott szoftverben azokat tartósan lekötött formában kell használni.

Ezek a szavak bosszantják leginkább azokat, akik előzőleg a szoftver idegen nyelvű változatát használták. Számukra ugyanis már konkretizálódott a széles jelentéstartalmú idegen szónak az adott összefüggésben érvényes egyetlen jelentése, és nem érdekli őket — talán észre sem veszik —, hogy a szó jelentése eredetileg sokkal tágabb volt. (Az is előfordul, hogy nem is olyan találó a széles spektrumú köznapi szó a jelölni kívánt fogalomra.)

Mi legyen az ollóval?

Ma már senkinek sem szokatlan, hogy „kivágás"-nak nevezik azt műveletet, amikor a program által előállított anyagból valamely részt úgy törlünk, hogy az később (máshol vagy ugyanott) újra felhasználható marad. A fogalom elfogadását segíthetik az általa kiváltott képzettársítások (ha ezek nem is mindig szakszerűek), és persze a jelen esetben a grafikus felületeken megjelenő, ollót ábrázoló piktogram is.

A „kivágás" elfogadása nélkül még zavaróbb lenne az az új műszó, hogy „vágólap". És hogy mit csinálunk azzal a valamivel, ami így a vágólapra került. Ez a valami, akárhányszor is helyezzük be valahova, mindaddig ott marad a vágólapon, amíg helyére (helyette) egy másik valamit nem teszünk. Mégsem „másolhatjuk" be a dokumentumba. A „másolás" ugyanis a kivágásnak egy speciális változatára van fenntartva: amikor a kivágás úgy történik, hogy a kivágott valami az eredeti helyén is ott marad.

A kialakított terminológia szerint ami a vágólapon van, azt a megfelelő helyre „beillesztjük", és nem „beszúrjuk". Miért? Csak. Így találták ki. Mára ezt mindenki megszokta, bár eleinte, különösen a hölgyfordítók, előszeretettel használták a „beillesztést" a sokkal durvább „beszúrás" helyett. Ki tudja, milyen tudatalatti képzettársításoktól vezéreltetve?

Tartsuk meg az idegen szót?

Az idegen szó átvétele sem megy olyan simán, mint gondolnánk. Felvetődik mindjárt az a kérdés, hogy képes-e nyelvünk befogadni az idegen szót, és hogy milyen mértékben lehetünk következetesek a használatában. Magyarrá tudjuk-e tenni például valamennyi érintett szövegrész nyelvezetét és a benne tükröződő gondolkodásmódot? A formális problémák sem elhanyagolhatók: ha megengedjük valamely idegen szó átvételét, zavarmentesen mondatokba tudjuk-e ágyazni minden esetben? Lehet-e igét rendelni hozzá? Használható-e jelzőként, igeként? Ragozható-e? Létezik-e előzménye, esetleg más értelme a magyarban, ahogy az a forrásnyelvben előfordulhat. Ugyanazt jelenti-e például magyarul a „szolid kvalitás", mint az angol „solid quality"? (Bizony nem!)

Hasonló problémák természetesen az általunk gyártott terminusoknál is felmerülhetnek. A fentebb említett szóválasztások ilyen szempontból szerencséseknek tekinthetők. Ők maguk olyan aktív igék, amelyek minden módban ragozhatók, könnyen képezhető belőlük főnév, melléknév vagy igenév. Számítástechnikai felhasználásuk sem okoz problémákat: nyugodtan használhatók leszűkített értelemben éppúgy, mint szélesebb körű, köznapi jelentésben, akárcsak eredeti angol megfelelőjük. Az ilyen megfeleltetés az ideális: az angol „insert" mindig lehet beszúrás, az angol „paste"-re pedig mindig megfelel a beillesztés. Hiszen a fordító még csak elkülönítve látja a szavakat, kifejezéseket, nem is tudhatja előre, milyen kombinációs lehetőségei vannak az adott terminusnak, milyen szörnyűségek jelennek majd meg a felhasználó képernyőjén.

Rendellenességgel születtek

A szoftver által generált egyes üzenetekről néha csak évek múltán derül ki, egy-egy ritkább hibajelenség felbukkanásakor, hogy mégsem olyan gömbölyűek, mint képzeltük. Az egyik üzenet „nappali mentési időt" emlegetett. Hogy ez tulajdonképpen fordítási hiba, csak azért derülhetett ki idejében, mert a tesztelési időszak éppen akkor volt, mikor áttértünk a nyári időszámításról a télire.

Persze nyilvánvaló fordítói hibák is akadnak szép számmal. De a legtöbb gyenge vagy kimondottan rossz eredmény abból származik, hogy nem ismerjük (nem is ismerhetjük eléggé) egy-egy kiragadott szövegrész rendeltetését. Az „Empty Folder" lehet egy tájékoztató állítás: ez egy üres mappa. De parancsként való használata is előfordulhat: töröld egy mappának a tartalmát. Ha ritkán is, de megeshet, hogy az angol szövegben ugyanazt a karakterláncot használják eltérő funkciók jelölésére, és ami jó az egyik helyen, az nem jó a másikon.

A tesztelés soha nem lehet elég mély, nem terjedhet ki minden lehetséges szituációra. Így történhetett meg, hogy a nyomtatási panelen nem „példányszám" lett a „copies" fordítása, hanem valami más; a bekezdés sorai között az egyik program magyar változata szerint szóközök vannak (mert az angolban „spaces" volt, de ott az mindenféle térközt is jelent), és hasonló, szépnek nem nevezhető megoldások. Az állapotsorban sem szabadna „Oszl 87"-nek állnia az angol „Col87" fordításaként, hiszen csak arról van szó, hogy a kurzor a 87. betűhelyen áll. Egy táblázatkészítéshez is értő szövegszerkesztőtől el lehetne várni, hogy meg tudja különböztetni a hasábot az oszloptól, akkor is, ha az angolban mindkettő „column". De szegény magyar honosító honnan bányássza elő az ehhez szükséges információkat?

Kikerülhető kátyúk

Üzenetekben nemegyszer találkozunk a magyar névelő kettős változatával: „a(z)". Ilyenkor egyértelmű, hogy az utána következő szó vagy kifejezés hogyan került oda: egy változó behelyettesítésével. Az ilyesmi nem helytelen, bár kissé csúnya. Használatát el tudjuk kerülni, ha például sikerül a behelyettesítésre kerülő szó elé beillesztünk egy jelzőt, amelynek a kezdőbetűje alapján már eldönthető, hogy a határozott névelő „a" vagy „az" változatára van-e szükség. Egy másik, nem mindig szerencsés megoldás lehet, hogy a névelőt elhagyjuk, esetleg úgy megforgatva a szórendet, hogy a behelyettesítendő szó a mondat elejére kerüljön.

Nehezebb a helyzet, ha a behelyettesítendő szó ragozott változatát kívánná meg a mondat. Ilyenkor is létezik megoldás, ha például tudjuk, hogy a behelyettesítésre kerülő szó mire vonatkozik, úgy is mondhatnánk, hogy „mi a típusa". Ezt a „típusazonosítót" beilleszthetjük a változóval jelölt szó után, ezt már ragozhatjuk, és egészen elfogadható mondatokat hozhatunk létre. Például a -ból/-ből rag kettősségét így lehet kikerülni a „Törli %s-ból?" helyett a „Törli a(z) %s nevű könyvtárból?" szerkezettel.

Az sem ritka eset, hogy egyes gépi úton előállított mondatok több változót is tartalmaznak. Ilyenkor az segíthet, ha az egyes változók számozással vagy más módon meg vannak különböztetve. Ekkor ugyanis a szórend módosításával gyakran megoldható a probléma.

De mi legyen, ha nem különbözteti meg? Ilyenkor jöhet a szófacsarás. A „Page %d of %d" kifejezésváznak például a legkülönfélébb magyar megvalósításával találkozhatunk:

„Oldalszám: 1/2",

„1. oldal, összesen: 2",

„1. oldal 2 közül" stb.

A behelyettesítésre kerülő dolgok „típusára" néha az angolban használt elöljárók jellegéből következtethetünk. Nézzük például a következő angol üzenetet:

„Accepting this item scheduled on ^0 at ^1 for ^2 will cause a schedule conflict."

Első megközelítésben egészen jónak tűnik egy ilyen magyar fordítás:

„^0 dátummal ^1 órakor ^2 ideig tartó tétel elfogadása időpontütközést eredményez."

Behelyettesítés után például ilyen mondatot kaphatunk:

„1996. 07. 15. dátummal 11:14 de. órakor 1 óra 15 perc ideig tartó tétel elfogadása időpontütközést eredményez."

Esetleg a következőképpen módosíthatjuk a fordítást, ha már pontosan tudjuk, mire vonatkozik az üzenet:

„Az ^0 dátumú, ^1 kezdetű, ^2 időtartamú esemény elfogadása..."

Ezek persze egyszerű példák, de sokkal nehezebb esetek is előfordulnak. Olykor a legjobb az lenne, ha a szoftver módosítását is el tudnánk érni, de a mi kedvünkért ezt aligha fogják megtenni. Csak legalább a később készülő szoftverek úgy készülnének, hogy a nyelvfüggetlenség előnyeit mi is maradéktalanul élvezhetnénk...

Hunglish üzenetek

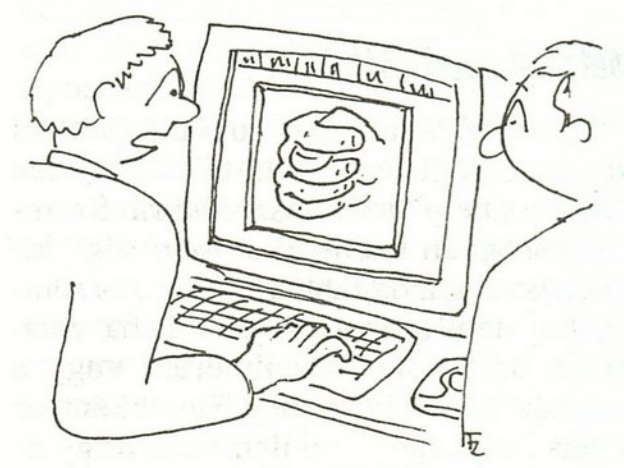
Kevert nyelvű üzeneteknek tulajdonképpen nem volna szabad előfordulniuk. Az élet azonban nem ilyen egyszerű. Most nem is az olyan durva hibákra gondolok, hogy a végső összeállításnál a nyelvet nem ismerő programozó könnyen belekever egy lengyel komponenst a magyar csomagba vagy fordítva. Kevert nyelvű megjelenítés adódhat egyszerűen abból is, hogy a korszerű szoftverek és a különböző alkalmazások képesek egymás objektumait is használni. Mikor az operációs rendszer kiszolgálja az alkalmazásokat, saját komponenseit mintegy kölcsönadja az alkalmazásoknak, ebből pedig nagyon bonyolult helyzetek adódhatnak.

Nem könnyű a felhasználó dolga akkor sem, ha takarékosságból a cég nem fordíttatja le a súgórendszert. Hiába érti a felhasználó, hogy az adott cél eléréséhez ezen és ezen a párbeszédpanelen az ilyen és ilyen feliratú kapcsolót be kellene jelölnie, nem biztos, hogy azonosítani tudja a megfelelő eszközöket. Olyan esetek is előfordulnak, hogy a súgó szövegét ugyan lefordították, de a szoftvertől függetlenül. Amelyik feliratra a súgó hivatkozik, az nem létezik a szoftverben, ami meg a szoftverben megvan, arról semmi tájékoztatást nem ad a súgó.

A fordítás közben keletkező sok „tücsök és bogár" egy része a béta-változatok tesztelése során (vagy előbb is) lelepleződik. Előfordulhat viszont, hogy azok elkerülik a figyelmet, vagy a kigyomláltak helyén újabbak jelennek meg, és ott lapulnak a közkézen forgó magyarított szoftverekben. Ha az olvasó találkozik ilyenekkel bármelyik cég termékében, ne hagyja annyiban. Nagy szolgálatot tesz mindnyájunknak, ha a hibát és jelentkezésének körülményeit leírja, és elküldi az Új Alaplap szerkesztőségébe, ahol Vargha Dénes vállalta az ilyen hibaüzenetek összegyűjtését és visszacsatolását az érintettekhez. Reméljük, hogy ezzel hozzájárulunk a jelenleginél „magyarul jobban beszélő" szoftverek készítéséhez.

Vaskó Tibor

IKONTECHNIKA



— Nemzetközi nyelven kommunikál...

A sztornó sztornózva

Terminológiai szópárbaj

Van egy aranyszabály, amely nemcsak a szoftverhonosításra igaz, hanem minden fordításra: egy szöveget valójában „fordítás" helyett megérteni és újra alkotni kell a másik nyelven. Ha ezt elfogadjuk, akkor természetesnek (sőt helyesnek) találjuk azt is, amikor a magyar terminológia egy mégoly ismert angol szónak sem a tükörfordítását követi.

Számítástechnikai környezetben mindenki ismeri azt a fogalmat, amelyet az angol a „Help" szóval jelöl. Szándékosan nem úgy írtam, hogy a „Help funkciót", mert az azt sugallhatná, mintha az angol kitüntetett lenne a nyelvek között. Nem kitüntetett, csak első. Angolimádat ide, sznobizmus oda, az angol számítástechnikai nyelv is csak ugyanazt csinálja, amit más nyelvek: fogalmakat jelöl szavakkal vagy kifejezésekkel. (Nagyon szépen fejezzük ki ezt a magyarban: „fogalma sincs róla".)

A fogalomnak és a fogalom nyelvi megjelenésének különbségét nem is mindig olyan könnyű felismerni. A különbségtételt nehezíti, hogy az emberi gondolkodás nem függetleníthető a nyelvtől, a dolgokra általában a jelzésül használt szó segítségével gondolunk. A szakemberek fejében a számítástechnikai fogalmak rendszerint nagyon szorosan kötődnek angol kifejezésmódjukhoz. Ez azonban nem jelenti azt, hogy ne lehetne a különböző nyelvekben különböző szavakkal jelölni ugyanazt a fogalmat. Még arra sincs szükség, hogy a célnyelvben választott szó és az angol eredeti elsődleges jelentésének bármi köze legyen egymáshoz.

Melyiket szeressem?

A „help" által jelölt funkció például valóban segítséget ad, tehát teljesen jogos a „help" szót használni rá. Szemléletességben azonban a „segítség" kifejezésnél gazdagabb a „súgó", és semmi baj nem származik abból, ha valakiben az iskolai osztályterem vagy a színház képét idézi fel. Egyébként az az érv is a „súgó" mellett szól, hogy ez a funkció munka közben áll rendelkezésünkre, és olyan média (a képernyő) közvetíti, amelyet magunk is használunk munka közben. Szembeállíthatjuk ezt a dokumentációval, amelynek megfelelője ugyanehhez a hasonlathoz közel maradva a színészeknek legépelve előre kiadott szöveg lehet. Ráadásul a súgó szó azt is kifejezi, hogy használata a munka tárgya szempontjából észrevehetetlen, nem zavarja meg annak integritását és folytonosságát. És végül egy formai érv: a képernyőn gyakran megjelenő szavak esetén óriási előnyt jelent, hogy milyen rövid a szó. (A súgó egyébként nem eredeti magyar találmány. A csehek is a színészeket előadás közben segítő súgó elnevezésével [nápoveda] jelölik a „help" funkciót.)

Mit fogadjunk el?

Hasonló dilemmát okoz az angol számítástechnikai nyelvben lépten-nyomon előforduló „support" szó. Ennek elfogadott magyar megfelelője — láss csodát — a tükörfordítása. Ezzel nem is lenne baj. Sok fordító azonban válogatás nélkül minden olyan esetben, amikor az angol szövegben a support szót látja, reflexszerűen már írja is a „támogatás" szót. Az eredmény így egy szürke, igénytelen magyar szöveg, amelyben minden második mondatban felbukkan a támogatás. Ha a fordító az angol szöveg ismerete nélkül magyarul fogalmazná meg a szöveget, magától talán azt írná, hogy „képes kezelni", „felismeri", „tartalmazza", így azonban nem a szöveg értelmét nézi, csak a szavakat. Az aranyszabály értelmében pedig ez csalhatatlan jele annak, hogy rossz a fordítás. Na persze, ha a fordító valóban nem érti a szöveget, akkor nem is tehet mást, csak kicseréli a szavakat, az elöljárókat ragokkal helyettesíti és kész. Csakhogy ebben nem sok a köszönet.

A magyar terminológiát rendszerint már a kész angol szövegek ismeretében kell kialakítni. Ez némi előnyt is jelent egyben, mert így lehetőségünk van kijavítani az angol terminológia néhány logikátlanságát. Ilyen logikátlanság származhat például a rokon értelmű szavak összekuszálódott jelentéseiből. Jellemző példa: a „choose" és a „select" szavakat az angol keverve használja, hol valamilyen parancs kiadására, hol bizonyos adatok megkülönböztetésére más adatoktól. Magyarban következetesebbek voltunk: az előbbire a „választ" szót használjuk, az utóbbira a „kijelöl" szót találtuk megfelelőnek.

Sikeres és sikertelen ötletek

Érdekes kérdés, hogy melyik szó bizonyul életképesnek, melyiket nem fogadja be a nyelv. Sajnos erre nincs szabály. Egy rossz szóról elég nagy biztonsággal meg lehet mondani, hogy rossz, de egy jónak tűnő szóról csak az élet tudja bebizonyítani, hogy megmarad-e.

Hasonlóképpen előfordul, hogy egy-egy „honosított" számítástechnikai kifejezés nem gyökerezik meg. A „Cancel" szónak van egy nagyon találó fordítása, amely minden igényt kielégíthetett volna. Rövid, pontosan azt írja le, amit a Cancel csinál, és mivel nem ige, kitűnő ellenpontja lehetett volna az OK gomb elnevezésének. Ez a szó a „sztornó", amely szintén idegen eredetű ugyan, de már meghonosodott a magyar nyelvben.

Már készen voltak a Microsoft első magyarra honosított szoftvereinek a bétái, bennük a sztornóval, amikor be kellett látnunk, hogy hiába küszködünk: ezt a szót nem fogják elfogadni a felhasználók. Még az az előnye sincs meg, hogy az angol terminológiát már ismerők „hozzák magukkal", mert nem büszkélkedhet előkelő angol származással. És hát, valljuk be, idegenes hangzásával nem igazán szép a szó. Pedig ha felvállaljuk egy új szó bevezetését, ilyesmit is mérlegelni kell. (A dolog pikantériája, hogy a béta-tesztelés eredményeként kapott negatív visszajelzések miatt mi megváltoztattuk a terminológiát a „mégsem" javára.

Egyes szoftverfejlesztők és honosítási műhelyek azonban addigra már átvették a sztornót, és végleges termékeikben is ezt használják.)

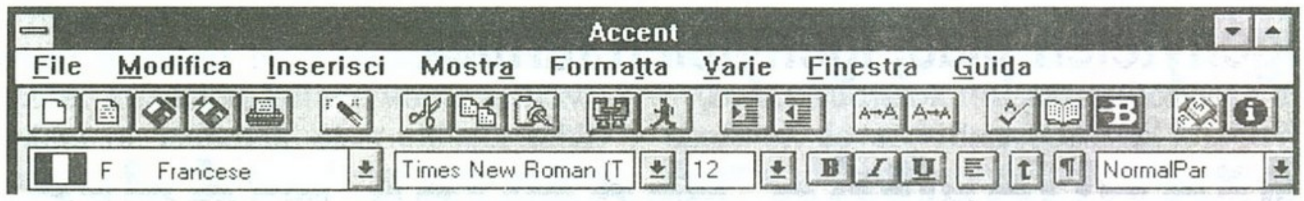
A lónak persze itt is van egy túlsó oldala: nem szabad mindenáron a Vereckénél Árpáddal bejött szavakat erőltetni. Minden nyelvben vannak jövevényszavak, amelyek megjelenése és integrálódása természetes és gyakran elkerülhetetlen jelenség. A protokoll például jövevényszó, de más értelemben vett közkeletűsége miatt azonnal elterjedt, és ma már aligha lehetne kiűzni a számítástechnikai terminológiából.

A file—fájl vitában annyira polarizálódtak a vélemények, hogy kár lenne az egészet elölről kezdeni. Én a fájl változatot sem kifogásolom, már csak azért sem, mert világunk egyre objektumorientáltabbá és dokumentumcentrikusabbá válik, így a „dokumentum" kezd a fájl helyére lépni a felhasználói programok kezelői felületén. A fájl egyre inkább szakszóvá válik, és angol eredete sokkal kevésbé zavaró. Egy dolog azonban továbbra is gondot okozhat a fájl használóinak: a szóvégi „jl" hangkapcsolat annyira szokatlan a magyarban, emiatt kedvünkre való magyarosított szó sosem lesz belőle, egyes ragos formáit (fájllal) kiejteni és leírni sem éppen könnyű. Igaz ugyan, hogy ezen problémákon a file írásmód sem segít, ezért én személy szerint jobban örültem volna annak — amire az angol ejtésmód elterjedtsége miatt viszont már szinte semmi esélyt látok —, hogy magyarosan filének is ejtsük (filével, filék, filéhez stb.).

A sikertelen újítás másik példája a szoftveres értelemben vett „driver" helyett javasolt „kezelőprogram". Talán azért nem terjedt el, mert túl általános. Újabban az „illesztőprogram" szót használjuk helyette, remélhetőleg nagyobb sikerrel. Egy biztos: nincs igazuk azoknak, akik szerint ezt a szót le sem kell fordítani, mert úgyis mindenki driverként ismeri. Azért tévednek, mert csak a szakmabeliek ismerik driverként. És nem a felhasználó van a számítástechnikáért, hanem fordítva.

A pontosság nem minden

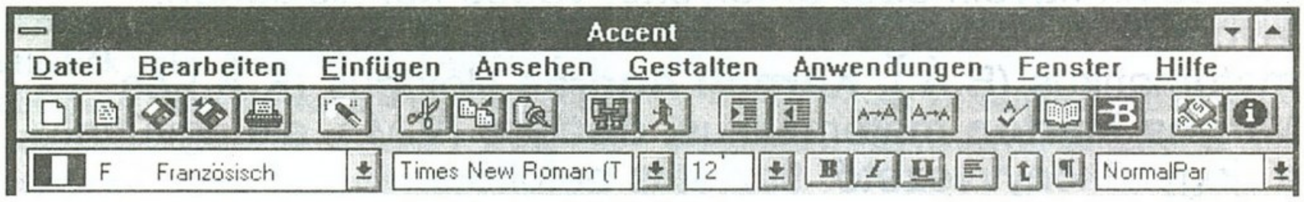
Közhely, hogy a terminológiának nem szabad pontatlannak lennie. Arra azonban kevesen gondolnak, hogy a pontosságot is túlzásba lehet vinni. Különösen szakmabeliektől kapunk olyan kritikákat, hogy egyik-másik magyar szó nem fedi pontosan azt a fogalmat, amire használjuk.



If you choose Russian, the Menu Bar will look like this:



If you choose German, the Menu Bar will look like this:



Előfordult már, hogy külön magyar megfelelőket találtunk ki két, első pillantásra hasonló, de valójában elkülöníthető dologra, az egyik mégsem honosodott meg. Így jártunk például az „ablak"—„párbeszédpanel" párossal. Erre a két fogalomra az angolnak is külön szava van („window", illetve „dialog box"), jól el is különíthetők egymástól az átméretezhetőség, alapszín, modalitás és egyéb jellemzők alapján. Első ránézésre viszont annyira hasonlóak (ráadásul az „ablak" a legelső fogalmak között van, mikor valaki megismerkedik az ablakozó technikát használó felhasználói felületekkel), bármit is választunk a dialog box megfelelőjeként, nehezen jön az ember szájára. Minduntalan „ablak" bukik ki helyette. Érdemes-e hát annyira erőltetni a másik kifejezést?

Érdekes példa az „italic" betűjellemző fordítása is a pontos terminológiára. A nyomdászok évszázadok óta kurzívnak nevezik, és szerintük itt is helyesebb lenne azt használni. A közérthetőség érdekében mi mégis megmaradtunk a „dőlt" szó mellett — és nem bántuk meg.

A pontos szakmai terminológiát néha be sem fogadja a köznyelv. A számítástechnikai alkalmazások felhasználói felületével többször van dolguk a titkárnőknek, adminisztrátoroknak, ügyvédeknek, kiskereskedőknek, mint a programozóknak vagy a rendszergazdáknak. Többségük nem ért a számítástechnikához, és nem is akar érteni. Nekik a szoftver minél könnyebb kezelhetősége és a szövegek megértése fontosabb a terminológiai finomságoknál.

Más a helyzet azoknak a fogalmaknak a fordításával, amelyek a rendszer legmélyén fordulnak elő. Ezeknél nemegyszer odáig kell menni az egzaktságban, hogy egy-egy fogalom angol elnevezése is egyértelműen azonosítható legyen. Ilyenkor az is megengedhető, hogy akár megmaradjon az eredeti angol elnevezés is (kernel, GDI, multicast).

Sziporkák

Vannak szavak, amelyek ötletesek, humorosak ugyan, de valamilyen fogyatékosságuk miatt nincs reális esély arra, hogy elterjedjenek, és nem is kerülnek bele a terminológiába. A laklap (home page), az emil (e-mail), a vakablak (clipboard) és társaik egy bizonyos körben sikeresek, de általánosan aligha terjednének el. (Az ilyen kifejezések valamiféle számítástechnikai szlengként a háttérben mégis megmaradnak.)

Tartok tőle, hogy legtöbb esetben éppen e szavak humorossága a legfőbb akadálya elterjedésüknek, mert bizonyos helyzetekben zavaró, hogy a szó komolytalannak tűnik. Máskor, például a „műszak" esetében — amely a „session" megfelelője lehetne — a magyar szóhoz tapadó asszociációk miatt nem lehet a szót terminusként használni. Hasonló a helyzet a „káptalan" szóval is, amely máskülönben kiváló fordítása lehetne a Windows vagy a Windows NT Registryjének, lévén ott minden megtalálható, amit a rendszerről tudni érdemes.

Varga Gábor

Igénytelen piac, igénytelen termék

Magyarítás magyar módra

Az alábbi beszélgetést inkább közös gondolkodásnak lehetne nevezni. Nem éles ellenvélemények csapnak össze — a résztvevők nézete az alapkérdésekben lényegében megegyezik. A három szakembernek (akik maguk is szereztek tapasztalatokat a szoftverek magyarítása terén) különösen azokat a gondolatait ajánljuk olvasóink figyelmébe, amelyeket végső konklúzióként levontak. A beszélgetés résztvevői: **Bedő Árpád** matematikus (B.Á.), **Kárpáti András** filológus (K.A.) és **Vargha Márton** matematikus, újságíró (V.M.), aki egyúttal a „jegyzőkönyvvezető" tisztét is betöltötte.

V.M.: Beszélgetésünk témája: a szoftver magyarítása. Ezt a problémát szerintem három oldalról érdemes megközelíteni. Az első a színvonal, a képernyőn és a kézikönyvekben megjelenő szövegek magyarsága. A második a terminológia. A harmadik meg az, hogy milyen viszony alakul ki a számítógép és a kezelője között, és hogyan tükröződik ez a szövegekben. Az alacsony színvonal okának én már régen az általános nyelvi igénytelenséget tartom. Emlékszem rá, az ÁSZSZ-nél (Államigazgatási Számítógépes Szolgálat — A szerk.) kezdetben még olyan sornyomtató volt, amelyen egyszerűen nem voltak ékezetes betűk. Megvettek egy számítógépet a Honeywelltől, súlyos milliókért, és nem követelték meg, hogy ékezetes betűszalagot adjanak hozzá. Pedig megtehették volna. De a műszakiaknak, úgy látszik, nem eléggé szívügyük a nyelv. Ez az igénytelenség már az általános iskolában elkezdődik, de még az egyetemekről is kerülnek ki végzett mérnökök, akiknek fogalmuk sincs például a helyesírásról. Egyszerűen föl sem fogják, mennyire fontos az emberek közötti kommunikáció eszköze. Akármerre nézünk, hivatalokban, irodákban, a közéletben, mindenütt igénytelenséget tapasztalunk.

B.Á.: Ne feledkezzünk meg azért a kérdés gazdasági oldaláról sem. Akármilyen széles körben használnak is majd például egy fejlesztő rendszert, ha nem magyar készítményről van szó, a külföldi gyártó első kérdése: „Mibe kerül?" Még mielőtt egyáltalán gondolkozni kezdene a magyarításon. Mindabból, ami a hibamentes működéshez kell, a minőségi ellenőrzés a legköltségesebb művelet. Vajon megtérül-e ez a költség? A szűk magyar piacon kicsi az esélye a megtérülésnek. Sok nagy szállító ezért dönt úgy, hogy nem is költ olyan luxusra, mint a magyarítás. Vagy ha igen, akkor legalább a minőségvizsgálat költségeit igyekszik megtakarítani.

V.M.: Az Adobéval kapcsolatos tapasztalataidra gondolsz?

B.Á.: Arra is. De ezek a megfontolások mindenkinél előjöttek, akivel egyáltalán kapcsolatban voltam. Az igaz, hogy éppen a betűkészítésben vetődik fel legvilágosabban a kérdés. Akik tervezik, metszik a betűket, nagyon jól tudják, hogy különböző népek különböző betűket használnak, de hiába. Még ha akarnák is, hogy a készletek teljesek legyenek, az anyagi megfontolások előbbre valók. Ha több változatot készítenek, több helyre el lehet adni. Van, ahová többet is egyszerre. De a mi változatunkat legtöbbször még csak el sem készítik.

Paradox módon nem is a szoftvernél, hanem a nyomtatóknál fizetünk rá a legtöbbet. Minden nyomtatóhoz megvásároltatják velünk a betűkészletet, benne van a nyomtató árában, de a legritkább eset, hogy a megvett készlet a magyar ékezetes betűket is tartalmazza. Kifizetjük, de használni sosem fogjuk.

K.A.: Éppen most volt egy ilyen telefonom. Egyik ismerősöm azzal hívott, hogy nem ért hozzá, de a nyomtató használati utasításában az van, hogy csak ezt meg ezt kell tenni, akkor majd jó lesz. Hiába teszi, a nyomtatója mégsem tud magyarul.

B.Á.: Hát persze. Azért is fizetünk, amit nem használunk. Aki magyar betűket akar, fizessen érte külön. Jöhetnének persze betű nélkül is a nyomtatók, olcsóbban. De hát ez is nevetséges volna.

V.M.: A programok kilencven százaléka nem is használja a nyomtató betűit. Windowsból grafikusan nyomtatnak, nem?

K.A.: Elvben beállítható, hogy a nyomtató betűit használja a Windows...

B.Á.: Jóval gyorsabb is lenne a nyomtatás, ha nem kellene mindent grafikusan nyomtatni.

K.A.: A betűk ügyében egyetértek, de Árpád, te a fejlesztő környezetet hoztad fel példának. Hogy azt nem fordítják le magyarra, szerintem nem olyan nagy baj. Viszonylag kevesen használják, szakmabeliek. Vannak azonban sok százezer példányban használt programok, amelyeket nem kérdés, hogy le kell-e fordítani. Operációs rendszert, szövegszerkesztőt, számolótáblát. Le is fordítják, de mivel a gazdasági kényszer dirigál, igyekeznek a lehető legkevesebb pénzből kihozni. A legkevesebb pénzből és a legrövidebb idő alatt. Valaki valahol kijelenti, hogy van rá hat hét, az alatt tessék lefordítani. Lehetetlen.

V.M.: Ez történt nemrég a Win95-tel.

K.A.: Például. Nem dolgoztam benne, de hallottam róla. Aki elvállalja, annak is tudnia kell, hogy azt lehetetlen rohammunkában jól megcsinálni. De gazdasági kényszer van, hát elvállalja. Ez a dolog tisztán gazdasági oldala. De mi miért vesszük meg? Ez már kulturális kérdés! Azt nem kérhetjük, hogy legyen egy bizottság, amely elbírálja a fordítást...

V.M.: Miért ne kérhetnénk, hogy alakítsanak erre egy bizottságot?

K.A.: Lehet, hogy kérhetjük, de az lenne a normális, ha a piac szűrné ki, ami rossz. Végül is, akinek egy csöpp nyelvérzéke van, annak elég belenézni egy ilyen termékbe, hogy lássa, hottentottául van, nem magyarul. Mégis megvesszük. És nem csak ezt vesszük meg. Minden vacakot, ruhát, cipőt, ehetetlen ennivalót, ihatatlan innivalót ránk lehet sózni — miért pont a szoftvert ne vennénk meg?

V.M.: Mindenütt el lehet adni mindent szemetet. Amerikában is eladják. Én is vettem Amerikában egydolláros kést, ma is azzal vágom a kenyeret. De például a hadsereg ott már nem ezt a kést veszi, hanem a jobb minőségűt. A

közigazgatás a nagy tömegű felvevő mindenütt a világon. Annak kellene előrevinnie Magyarországon is a dolgokat, mégis ők járnak az élen az igénytelenségben.

B.Á.: Számomra örök rejtély, hogy a közigazgatáshoz, a belügyhöz tartozó hivatalokban, amelyeknél hihetetlen mennyiségű gép, nyomtató van, ahol rengeteg az adat, az irat, mennyire nem használják ki a lehetőségeiket. Hogyhogy ez a nagy mennyiség nem tud semmit kikényszeríteni? A piacon pedig ez már olyan vásárlóerő, aminek lehetne kényszerítő ereje.

K.A.: Van egy másik természetes közeg is, amely ki tudná kényszeríteni a minőségi árut: az oktatás. Magyarul folyik az oktatás, oda nem volna szabad akármit bevinni. Csak az a baj, hogy nálunk nagyon gyenge és kicsi az oktatási piac. Gyakorlatilag nem is létezik.

V.M.: Illetve annyira van, hogy kezet csókolnak mindenkinek, aki ad egy gépet, és eszükbe sem jut a fogát megnézni.

B.Á.: A Microsoft — amennyire bírja — elárasztja az iskolákat is, hogy hozzászokjanak az ő termékeihez.

K.A.: Lehet-e ebben a helyzetben valamit csinálni? Reménytelennek látom, hogy egy magyar céget, amely elvállalja egy szoftver magyarítását, rá lehetne kényszeríteni arra, hogy minőséget gyártson.

B.Á.: Szerintem volna rá mód. A számítástechnika egyre inkább átszövi az életet, és az emberek beszédét is. Magam azt tapasztaltam, hogy a programok erősen befolyásolják a szóhasználatot. Ott vannak a főiskolák, egyetemek, különféle oktatási intézmények, ahol előadnak emberek, akik biztosan magyarul beszélnek, nem angolul. Használnak egy nyelvet, amely szükségképpen el kell, hogy térjen a programok nyelvétől, hiszen azoknak a nyelve sem egységes.

V.M.: Te most átugrottál a szaknyelvi kérdésre.

K.A.: Az teljesen más ügy.

V.M.: Ott én elfogadhatónak tartom az idegen kifejezések használatát. Kialakul egy konyhanyelv, amelyet néhány tízezer ember fog használni. Baj akkor van, ha ez a nyelv átszüremkedik a hétköznapi életbe, megjelenik a szakértő és az egyszerű felhasználó közötti kapcsolatban is. A számítógép ugyanolyan eszközzé válik, mint a tűzhely vagy a telefon. Az igazi kérdés az, hogy milyen nyelvet kapnak a géptől azok, akik a mindennapi életben használják, milyen nyelven kommunikálnak egymással.

Ez a technológia az utolsó harminc évben áramlott be a társadalomba. A kezdet kezdetén kellett volna már vigyázni rá, azoknak, akik először találkoztak vele. Most is az első találkozásra kell sokkal jobban odafigyelni. Ha erre gondoltál, amikor az egyetemről beszéltél, akkor egyetértek, hiszen ami az egyetemeken megjelenik, az egyenesen megy át a társadalomba.

B.Á.: Beszéljünk még a bizottságról. Hasonló ez ahhoz, ahogy az építés szabályozva van. Nem lehet akármilyen homlokzatú házat építeni Budapesten. Az építésznek van ugyan tervezői jogosítványa, de vannak fórumok, azok felülbírálhatják. Hiába önálló szellemi alkotása a terv a tervezőnek, átalakíttathatják vele. A szoftvernél nincs ilyen lehetőség. Készít valaki egy magyar szövegszerkesztő-változatot, amelyet százezrek fognak használni, és azt nem kell sehol bemutatni, nem kell sehol minőségvizsgálatnak alávetni. Nyugodtan eladhatja, és meg is veszik — no nem azért, mert rácsodálkoznak, hogy milyen szépen beszél magyarul, hanem mert használniuk kell. Bizony nem ártana a piacra vitelt minőségellenőrzéshez kötni.

V.M.: A Magyarországon forgalomba hozható termékekhez kötelező magyar nyelvű használati utasítást adni. Villanyborotvához adnak, szoftverhez nem. Programot nyugodtan be lehet hozni Magyarországra magyar nyelvű használati utasítás nélkül. Itt kellene megfogni a dolgot. Ez lenne az első lépés, a jogi szabályozás. Azután már a többi szabályt is be lehetne tartatni, ha precedens lenne, hogy a szoftveren is számon kérik ezt az előírást.

K.A.: Ha van is a szoftverhez valami magyar leírás, kézikönyv, menü, vagy a help fordítása, az hiányos és igénytelen, ráadásul gyakran félrevezető információkat ad. A nyelvezetéről meg jobb nem beszélni. Pedig ha magyartalan, akkor rontani fogja a felhasználó nyelvérzékét.

B.Á.: Tudjátok, mi itt a különbség a szoftver és a villanyborotva között? Amit a villanyborotva mellé adnak, az a szöveg jóval nehezebben épül bele a kultúrába. Amit viszont az emberek a programmal hazavisznek, az nem egyszeri elolvasásra való. Ezért sokkal veszélyesebb terület.

V.M.: Tehát egyetértetek, hogy jogszabályra van szükség?

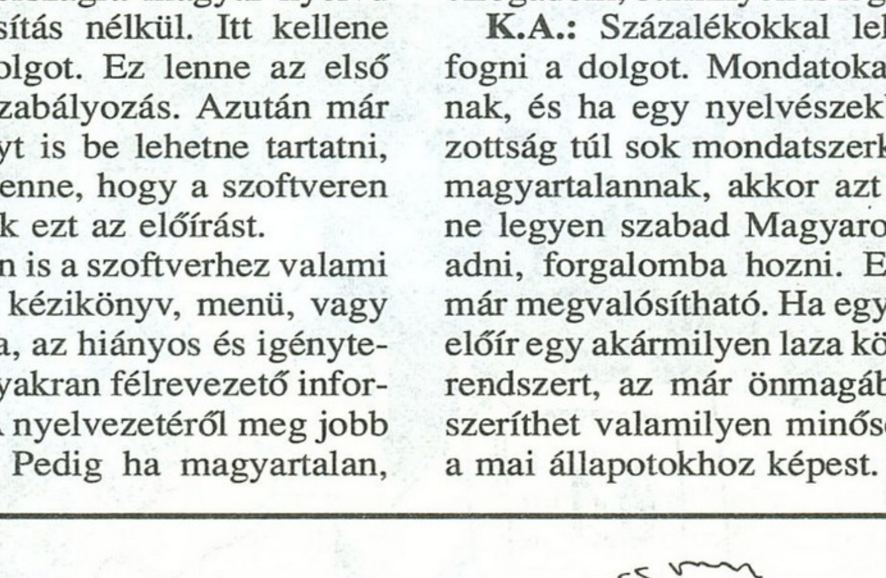
K.A.: Ha ezt el lehetne érni, akkor azonnal anyagi érdekük is fűződne hozzá a cégeknek, hogy odafigyeljenek, hogy áldozzanak a minőségre. Ha tudja, hogy megbukhat a hat hét alatt összehozott változattal, akkor másképp indul neki.

V.M.: Jó, de ki csinálná a minőségvizsgálatot? Itt nyelvről és ízlésről van szó.

K.A.: Ezért vagyok én szkeptikus. Nem lehet megtalálni azokat a szempontokat, amelyek alapján ez megvalósítható. Emlékszel? Vitatkoztunk mi már stíluson. Én azt mondtam, főnevekkel fejezzük ki, amit lehet, te pedig az igei kifejezések, az utasítások mellett kardoskodtál. Ez két eltérő álláspont. Ki döntheti el, hogy melyik érvényesüljön?

B.Á.: Ami elér egy adott igényességi szintet, ami eleget tesz bizonyos alapvető minőségi követelményeknek, azt elfogadom, bármilyen is legyen a stílus.

K.A.: Százalékokkal lehetne megfogni a dolgot. Mondatokat vizsgáljanak, és ha egy nyelvészekből álló bizottság túl sok mondatszerkezetet talál magyartalannak, akkor azt a szoftvert ne legyen szabad Magyarországon eladni, forgalomba hozni. Ez így talán már megvalósítható. Ha egy jogszabály előír egy akármilyen laza követelményrendszert, az már önmagában kikényszeríthet valamilyen minőségi javulást a mai állapotokhoz képest.



— Én nem ilyen fordítóprogramra gondoltam!

B.Á.: Kulturális szempontból azért is rendkívül indokolt ebbe az irányba haladni, mert elfogadhatatlan, hogy piaci érdekek határozzanak meg alapvető kulturális változásokat. Méghozzá nem is Magyarországon belüli piaci érdekek. A hatást persze nem lehet kiküszöbölni, de nem is arról van szó...

V.M.: A szoftver hihetetlen mértékben képes terjeszteni az idegen és idegenes gondolkodásmódot. Ha nem vigyázunk, előbb-utóbb eltűnik a sajátunk, és a gyerekeink gondolkodni is angolul fognak.

B.Á.: Nehéz például szabadulni attól, hogy angolul állapotokkal fejezik ki azt, amit magyarul inkább folyamatként írnánk le. Ez alapvető különbség a két nyelv között. Akinek nincs elég ideje, szakmányban fordít szövegeket, az óhatatlanul megtartja az állapotszemléletet, pedig nem volna kötelező. Én magam is többször beleestem ebbe a hibába.

K.A.: Én is csináltam már korrekt állapotszemléletű fordítást. Amikor sietnem kellett... Egyébként ami a kulturális oldalt illeti, szerintem legjobb lenne kiragadni az informatikát a számítástechnikusok kezéből, amíg nem késő. Az lenne a csodálatos, ha a gimnáziumokban, iskolákban ezt nem a matektanár tanítaná, ő elég, ha programozni megtanítja a gyerekeket. Hogy mi az informatika, az informatikai kultúra, azt inkább a magyartanárnak kellene tanítania.

B.Á.: Ti tudtok az Interneten ékezetes betűkkel levelezni?

K.A.: Nem.

V.M.: Én CompuServe-ön levelezek, ott lehet. De lehet a Zöldpókon, a környezetvédők hálózatán is, Emillel.

K.A.: Miért nem adnak az internetes kapcsolatszolgáltatók mindenkinek olyan szoftvert, amelyik ismeri a magyar ábécét? Azt kellene elérni, hogy az Internet-szolgáltatók minden előfizetőt ellássanak magyarított, szépen magyarított böngészővel és levelezővel. Most mindent angolul adnak, és az emberek ebből sajátítják el a hálózati kultúrát. Terjednek az angol szavakból gyártott magyar igék, és ez már óhatatlanul beleépül a kultúrába.

V.M.: Zsadányi Pál minden adandó alkalommal rákérdez erre, eddig eredménytelenül. A magyar nyelv háttérbe szorulásához képest tulajdonképpen apróság, hogy nem lehet ékezetes betűket használni.

K.A.: Az viszont nagy baj, ha a keretprogram, a böngésző nincs megcsinálva magyarul. Ezeknél is meg lehet követelni azt, amiről az előbb az alkalmazásokkal kapcsolatban beszéltünk. Elég lenne kimondani, hogy a közigazgatásban, az iskolában csak olyan programok használhatók, amelyek magyarul közlekednek a felhasználóval. Amit feltesz a gépére a közalkalmazott, a köztisztviselő, annak magyarul kell lennie. Helyes magyarsággal.

B.Á.: Vélhetően a közigazgatás a legnagyobb piac, könnyen egységesíthető is. Ott kellene elérni, hogy jó minőségben magyarított programokat vásároljanak. Kérdés azonban, hogy ki állja ennek a költségeit. Csakis nekünk, magyaroknak érdekünk, hogy az informatika magyarul legyen, hogy lehessen magyarul levelezni, hogy gondolkodásunk ne idomuljon teljesen az angolszászhoz?

V.M.: Támogassák az adófizetők?

B.Á.: Miért ne? A korrekt magyarítást támogathatná az állam. Követelőzni is lehet az amerikai cégeknél, de többet ér, ha együttműködést ajánlunk fel. Ha a jogi szabályozás elkészül, ha a legnagyobb vásárló elutasítja a magyartalan szoftvereket, akkor tárgyalási pozíciónk is megerősödik. És persze alapos kritikára, kritikai szellemre volna szükség. Én például soha semmilyen nyilvános kritikát nem kaptam arra, amit ezen a területen csináltam. Például a Macintosh rendszerre vagy a kézikönyvekre.

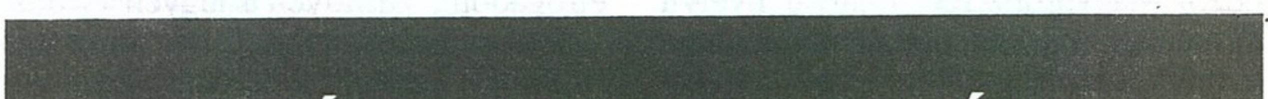
V.M.: Amikor a Windows 3.1 hivatalos magyar változata megjelent, én sorozatban bíráltam, de nem sikerült kialakítani igazán alkotó, kritikus légkört. Mindenki csak magában hőbörög — azután lassan hozzászokik a legrosszabb megoldásokhoz is.

K.A.: Az igazi szoftverkritika és a szoftvermagyarítás érdemi kritikája egészen különleges műfaj. Ennek talán még a szabályait is meg kellene alkotni. És igényes vásárló is kell a jó kritikához, aki odafigyel rá. Ha a kritikának meglesz a megfelelő súlya, akkor a gyártó és az eladó is kénytelen lesz majd odafigyelni.

V.M.: És a terminológiával, a fordításokkal mi a helyzet?

B.Á.: Úgy látom, ez teljesen esetleges, véletlenszerű. Kitalálunk valamit, azután vagy elterjed, vagy nem. Például a telepítés, az bejött, elfogadta a szakma. Lehet a szavakon lovagolni, de ma már úgy látom, hogy nem érdemes. A bajok sokkal mélyebbek, nem azon múlik, hogy file-t, fájlt, iratot vagy dokumentumot írunk a menübe.

K.A.: Szerintem sem. Amiről eddig szó volt, a mondatszerkezetek, a gondolatok magyarsága sokkal fontosabb, mint a szavaké. Ha létrejönne — bár létrejönne! — egy jó jogi szabályozás, ha a nagy megrendelők visszadobnák a csúnyát és a rosszat, akkor a terminológia kérdése is megoldódna.



EBBŐL KELL KIHOZNI A LEGTÖBBET

A regionális elv árnyékában

Kódok háborúja

Mikroszámítógép Magazinból 1990 júniusában Alaplappá átvedlett folyóiratunk legelső „hónap témája", a MAGZARUL BESYELUNK címet viselte, és azóta is rendszeresen foglalkoztunk a magyar nyelv és a számítástechnika kapcsolatával, különösen a karakterkódok rendszerével. Mostani összeállításunkban is célszerűnek láttunk naprakész summázatot adni ez utóbbi témáról.

Az 1992-ben készült és 1993. március 1-jén életbe lépett magyar szabvány (MSZ 7795-3:1992) kétféle kódtáblát szentesít. A magyar ékezetes betűk kódját az egyik kódtábla a számítógépes cégek „Latin-2" vagy „852" néven ismertté vált „házi szabványából" veszi át (hivatkozás nélkül), a másikat az ISO 8859-2-ből. Ez az utóbbi tulajdonképpen megegyezik a Windows 1250-es kódtáblájával.

Az adatbevitel szempontjából elsődlegesen annak a kódkészletnek van jelentősége, amely karakteres formában, közvetlenül a billentyűzetről kerül a gépbe. Itt ugyanis óriási káoszt okozhat, ha különböző kódrendszerek szerint készített adatok keverednek össze, és mondjuk egy adatbázisba „kakukktojások" kerülnek.

Közvetlen karakteres inputként a Windows *grafikus* felület kódkészlete kevésbé problematikus, érdemes viszont közelebbről megnézni a *karakteres* felületű adatkezelés problémáit, mivel a 852-es által kavart vihar máig sem csitult el teljesen. Egymás mellett élnek különböző kódkészlettel készült állományok, különböző kódokat használó rendszerek, sőt olyan rendszerek is, amelyekben keveredik többféle kódkészlet.

Ki a legény a gáton?

Piaci szempontok diktálták a hardver- és szoftvergyártó cégeknek azt a törekvését, hogy hallgatólagos szabványként elfogadtassák a 852-es kódtáblát. Ezzel teljesen megzavarták, szétzilálták a nagy nehezen már az egységesülés felé elindult folyamatot.

Előzőleg a hazai szakemberek már belátták az „ahány ház, annyi kódkiosztás" gyakorlatának tarthatatlanságát, és többnapos vita alapján megszületett az a kódkiosztási ajánlás, amely CWI-kód néven hamarosan országszerte elterjedt, és spontán módon egységesedni kezdett a magyar PC-s társadalomban a kódhasználat. (Nem a hivatalos úton, mert az 1984 óta érvényes magyar szabvány tulajdonképpen nem érdekelte a PC-seket — az MSZ 7795-3:1984 csupán a PC-ktől idegen EBCDIC kódoknak egy KGST-mutációját szabályozta.) Az elkészült CWI kódtáblázat nevét később CWI-1-re módosították, mert közben elkészült a CWI-2 is, egyetlen apró változtatással: a nagy hosszú „Í" ASCII 140-es helyett a 141-es kódot kapta — szélesebb körben inkább ez terjedt el.

A CWI-kódnak egyetlen komoly vetélytársa volt, a Ventura kódkiosztása, amely az ASCII keretrajzoló karaktereinek rovására adott helyet a magyar ékezetes betűknek, ezért a számítástechnikusok nem is szerették, hiszen kedvenc játékszerüket rabolta el.

...Nos, a CWI vonalon egységesedő magyar kódkiosztás elterjedésének folyamatába robbant bele a 852-es elfogadtatására irányuló kampány, ami a 80-as évek végén máig is tartó ellenkezést váltott ki a hazai számítógépes társadalomban. (Lásd erről részletesebben az Alaplap 1990/6., 1992/5., az Új Alaplap 1994/3. számát.)

A nemzetközi porond

A Nemzetközi Szabványügyi Szervezetben (ISO) már a 80-as évek eleje óta folytak a számítógépes kódok nemzetközi szabványosításával kapcsolatos munkálatok. Ezeknek a szabványosítási törekvéseknek azonban alapvetően mások voltak a szempontjai: olyan kódtáblák elkészítését tartották fontosnak, amelyekben helyet kapnak valamennyi latin betűs írást használó nyelv karakterei. És bár az egész világra kiterjedően ez akkor még nem volt reális célkitűzés, legalább Európa nyelveivel kapcsolatban törekedtek a teljességre.

El kell ismerni, az ISO-nak sem volt egyszerű dolga az új kódtáblák kidolgozásakor. A különleges, „módosított latin" karakterek a legtöbb gondot a walesi (welsh), valamint a portugál és a szlovák nyelv esetében jelentették. A kelta hagyományaikat ma is őrző walesiek a 26 betűből álló alapkészleten kívül 24 módosított latin betűt is használnak, a nagybetűkkel együtt tehát csak az ő igényeik kielégítésére 2 x 24 karakterrel kellene kiegészíteni a kódkészletet. Nem csekélység a portugálok és a szlovákok 2 x 17, a grönlandiak és a lappok 2 x 15 karakterigénye sem, még ha sok is az átfedés az egyes karakterkészletek között. A kizárólag a saját nyelvükben használatos legtöbb betűvel a lengyelek dicsekedhetnek: 2 x 8-cal; viszont nagyon nagy (több, mint 50) azoknak a betűknek a száma, amelyeket csak egy vagy két nyelvben használnak.

Nyilvánvaló volt, hogy minden karaktert nem lehet elhelyezni egyetlen 8-bites kódtábla keretei között, főleg ha a vezérlőjeleknek, a számoknak, az írásjeleknek, a dobozkaraktereknek, a matematikai műveleti és relációjeleknek, a pénznemeknek, a matematikai és fizikai szövegben gyakran használatos görög betűknek stb. is helyet akarunk szorítani, ahogyan azt a nálunk is jól ismert „kiterjesztett ASCII" megkísérelte.

A sokféle szempont összeegyeztetéséből, bár nem éppen szép, és nem is nagyon következetes, de elfogadható kompromisszum született. A nyelvek közül kiemelten kezelték a francia, az olasz és a spanyol nyelv igényeit. Rendkívül mostohán bántak viszont a nagybetűkkel: az ékezetes (mellékjeles) betűk többségének nincs meg a nagybetűs párja. Dehát tudjuk, hogy a „hazai" franciák nem is használják az ékezetes betűk nagybetűs alakját, csak a kanadai franciák.

Táguló világegyetem

Eleinte az ISO-ban is a kiterjesztett ASCII kódkiosztásából indultak ki, hamarosan azonban gyökeresen szakítottak azzal az összevissza kódkiosztással, amely még a 852-est és társait is jellemezte. Kompromisszumok helyett hozzáláttak kódtáblacsaládok kialakításához. Az első 127 kódhelyen ezek a kódtáblák szigorúan betartották azt a 7

bites sorrendet, amelyet az 1973-es ISO 646 szabvány rögzített. A 8 bites kiterjesztésben többnyire a kódlapok cserélgetésének az ISO 2022 szabványban szabályozott elveihez igazodtak (ami jól megfelelt a kódtáblacsaládok koncepciójának), de figyelembe vették a 7 bites és 8 bites szövegátvitelre vonatkozó S.61 és S.100 jelzetű CCITT-ajánlásokat is.

1983-ra készült el az első olyan szabvány, az ISO 6937, amely már a kisbetű—nagybetű párok elhelyezésében is igyekezett következetes lenni (mindkettő legyen meg, egymástól ugyanolyan távolságban, mint a kódtábla első felében a kis- és nagybetűk). Ilyen értelemben ez volt az előfutára az „ANSI szabvány"-ként ismertté vált kódkiosztásnak, amelynek elveihez később a Windows is igazodott. (ANSI: American National Standards Institute. Saját nevét az ANSI csak ehhez a kódkészlethez, és az ebből származtatott további kódkészletekhez adta.)

Ha nem is terjedt el széleskörűen az ISO 6937, de megvetette az alapját a körültekintőbb munkálatoknak, például a szükséges és elégséges kódkészletek kidolgozásának. Ekkortól kezdték komolyan feltérképezni és rendszerbe foglalni a lehetséges igényeket, például azt is, hogy mennyi hely kellene, ha a világ valamennyi latin betűs nyelvét figyelembe vennék, mit jelentene, ha hozzávennék azokat a speciális karaktereket is, amelyek a nem latin betűalakokat használó nyelvek egyértelmű oda-vissza transzliterálásához szükségesek. Végső soron a 90-es évekre beérett Unicode-koncepció is az ISO szakembereinek ezekből a korai hagyományaiból merített, még munkaanyagaikat is felhasználta.

Latin-2 kontra CWI

Az ISO előmunkálatait a számítógépes cégek is felhasználták, csupán jobban előtérbe helyezték azt a számukra elsődleges szempontot, hogy az új kódkészletek regionális elvek alapján készüljenek. A mi esetünkben ez azt jelentette, hogy a magyar nyelv karakterei ne a nyugati nyelvek karaktereivel együtt legyenek használhatók, hanem a keleti latin betűs nyelvek (lengyel, szlovák, román stb.) speciális karaktereivel.

E regionális szabályozás jegyében született meg a „852"-esként ismert kódtábla, amely lényegesen eltért az itthon már-már egyértelműen elfogadott és számos adatbázisban is alkalmazott CWI-kódtól. A 852-es szerint olvasva az ékezetes karakterek CWI-kódja helyén nyolc betűnél (Á, Í, Ó, ő, Ő, Ú, ű, Ű) lengyel és szlovák karakterek jelennek meg a képernyőn; ha pedig ASCII-ban vagy CWI-ben nézzük a 852-es ékezetes karaktereit, akkor az újonnan igénybe vett kódhelyeken jobbára dobozkaraktereket, görög betűket, matematikai jeleket látunk feltűnni a magyar szöveg közben.

Az újabban Magyar Szabványügyi Testület névre átkeresztelt intézmény, bár minimális változtatással, hivatalosan is elfogadta a 852-es javaslatot. A magyar karakterek elhelyezésével kapcsolatban ezt tette szabvánnyá 1993-tól, más karakterek tekintetében azonban szabad kezet adott a szoftverkészítőknek tetszőleges két- vagy többnyelvű kódkészlet alkalmazására.

A nagy kérdés természetesen az, hogy az összhang mennyiben nélkülözhető, vagy hogyan biztosítható ilyen igények felmerülésekor. Lehet-e megfelelő szoftvertámogatást találni, mondjuk a többnyelvű adatbázisok fejlesztéséhez, vagy a nyomdákkal való együttműködésben a többnyelvű kiadványok elkészítéséhez?

Nem lehetne békésen megoldani?

Ha a szoftverkészítő cégek egy picikét is törekednének a helyi igények kielégítésére, és nem saját hegemóniájuk megteremtése lenne a legfőbb céljuk, akkor a CWI és saját kódkészletük között is biztosítanák a kétirányú közlekedést. Könnyen megszüntethető lenne például az az anomália, hogy amikor CWI-ben írt szöveget Windows-zal beolvassuk (mondjuk azért, mert így akarjuk kinyomtatni), akkor, ha konvertálást kérünk, a fenti nyolc ékezetes karakter helyén zavaróan idegen karakterek jelennek meg a magyar szövegben. (Feltehetőleg elég könnyen meg lehetne oldani, hogy a jelenlegi „bedrótozott" 852-es kódtábla mellett másféle kódtáblát is választhasson a felhasználó. Mondjuk, azzal a megszorítással, hogy a kívánt betűalakok csak sorrendjükben térhetnek el azoktól, amelyeket a rendszer jelenleg a „bedrótozott" kódtábla alapján elő tud venni. Számítástechnikailag talán nem a legbonyolultabb probléma.)

Közben tovább éltek és szaporodtak a regionális elven alapuló kódtáblák. Az ISO is segített abban, hogy csoportokba foglalják a karaktereket, és kódkiosztásban egymáshoz többé-kevésbé hasonló, viszonylag széleskörűen használható kódtáblák készüljenek a kiterjesztett ASCII család folytatásaként. Szép gyűjtemény található ezekből a Microsoft anyagaiban az 1000-nél kisebb sorszámú kódlapok között. Ide tartozik többek között a sokat emlegetett 852-es kódlap is, amelynek nyugat-európai megfelelője a 850-es, az USA-beli pedig a 437-es.

Átgondolt, mély koncepciót nehéz lenne felfedezni akár a felvett karakterekben, akár azok elhelyezésének módjában. Inkább adhoc fejlesztések eredményének látszanak ezek a kódtáblák, ahol a legfőbb szempont az volt, hogy valahogyan minden felvetődött igényt ki tudjanak elégíteni. A Latin-2 kódtábla után (amelyet néha szlávnak neveznek, máskor közép-európainak) hamarosan megjelent a kanadai francia (MS számozás szerint: 863), majd főleg a skandináv igények kielégítésére kidolgozott „északi" (865) és a balti (775). Jellemző, hogy külön kódlap készült a portugál (860), az izlandi (861) és a török (857) nyelv számára is, esetenként csupán azért, mert egy-egy betű elhelyezéséről előzőleg elfelejtkeztek.

Ezekkel a fejlesztésekkel párhuzamosan, hasonló elvek alapján készült el a cirill (855 és 866), a görög (737 és 851), de még a héber (862) és az arab (864) kódtábla, sőt több távol-keleti változat is. A kódtábla alsó felében azonban — ezt fontos megjegyezni — ezekben is megmaradtak a szűkebb értelemben vett ASCII karakterek. A latintól alapvetően eltérő betűkészletet használó írásrendszerek esetében ez biztosította a velük készült szövegeknek angol szövegekkel vegyes használatát.

A Windows kódnyelve

Az ANSI először saját kebelében, majd az ISO-val együttműködve kezdte szisztematikusan rendbe tenni az addig globálisan nem eléggé átgondolt fejlesztéseket. A kódtábla alsó felét továbbra sem bántották, a 7 bites kiosztást szabályozó ISO 646 szabvány minden további fejlesztésnek támasza és talpköve maradt. Kihagyták viszont (vagy külön táblázatban helyezték el) a „rendszeridegen" jelek java részét: a dobozkaraktereket, a görög betűket és az aritmetikainál magasabb fokú matematikai jeleket.

Az ANSI-fejlesztés korai szakaszának eredményeit vette át az „1252 Windows Latin-1" kódtáblája, s a 852 mintájára ebből alakította ki a közép-európainak is nevezett „1250 Windows Latin-2" kódtáblát. Ebben egyes francia, olasz, spanyol, portugál, svéd, norvég, dán, izlandi betűk helyett, ill. tartalék betűhelyekre tettek be lengyel, cseh,

szlovák, horvát, magyar és román betűket, véleményem szerint egészen ügyesen. Némi vacillálás egyedül abban volt, hogy a kis és nagy magyar „Ő" betűt melyik betű helyére tegyék be: a portugálban (ésésztben) használt hullámos ékezetű „Õ" betű, vagy a franciában, portugálban és sok kisebb nyelvben használatos „háztetős" ékezetű „Ô" betű helyére, az „Ű"-vel analóg módon.

Végül az utóbbi mellett döntöttek, bár ebben tapasztalható némi ingadozás az egyes rendszerek között. Biztosan mindenki bosszankodott már amiatt, hogy bizonyos betűtípusok a magyarított változatban sem ismerik a magyar „ő" betűt. Nos, árulkodó módon ilyenkor az a betűalak jelenik meg, amelyet a magyarnak helyettesítenie kellene. A másik választás mellett is komoly érvek szóltak volna. Például az, hogy a háztetős ékezetű „O-származék" (Ô) nemcsak hogy sokkal több nyelvben fordul elő, de köztük van a szlovák nyelv is. Ennek a betűnek a kihagyásával tehát a szlovákok számára a betűkészlet nem lesz teljes, ami koncepcionális hibát sejtet. Vagy talán egy közép-európai szövegekben is fontos betűalakot akartak megőrizni? De ha itt igen, akkor más esetekben miért nem?

Mondd, mi fáj?

Végeredményben a grafikus környezetben megjelenő kódok belső tárolási módjával ritkábban kerül összeütközésbe a magyar felhasználó, mint amikor karakteres üzemmódot használ. A Windows-zal sem a karakterei, a kódkiosztása miatt szokott problémánk lenni, sokkal inkább a különböző szoftverek eltérő, következetlen működése miatt. Például azért, mert a Microsoft elfelejtett arról gondoskodni, hogy a magyarított, ékezetes TrueType fontok más környezetben, nem magyar Windows alatt is működjenek. Ráadásul ezt a hibát átvitték a Win95-re is! (Mellesleg a „páneurópai" Win95-ben nem lehetne esetleg Magyarországot is szerepeltetni?)

Egyébként is rengeteg következetlenség észlelhető abban, hogy a Win95 háromféle változata (USA, magyar, páneurópai) mikor hogyan viselkedik. Külön változtatható a billentyűkiosztás, külön a rendszer nyelve, de hogy mikor melyik beállítás élvez prioritást és érvényteleníti a másikat, azt még kikísérletezni is nehéz. A Súgó pedig bölcsen hallgat, mihelyst kényesebb kérdést teszünk föl neki.

Még egyetlen rendszeren belül is vannak következetlenségek. Az Office-on belül például „be van betonozva", hogy mikor mit hívjon meg. A nyelvválasztást már nem támogatja — pedig magát a beállítást látszólag még elfogadja.

A billentyűzet kezelésével is vannak problémák. Előfordul, hogy a magyarított Office szövegszerkesztője nem ismeri fel a billentyűzet beállítását, és olyankor is saját „forrógombos" funkcióival reagál a jobb oldali („szürke") Alt-tal együtt lenyomott gombra, mikor a kiszorított USA-jeleket szeretnénk előhívni.

„Éles", üzemszerű felhasználásban kellemetlen következményekkel járhat az olyan rendellenesség, amikor a képernyőn más jelenik meg, mint amit a nyomdai levilágító rendszer majd látni fog. Elsősorban olyan „többszörös kódok" esetében szokott ez megtörténni, amelyek többféle kódkiosztásnak is igyekeznek eleget tenni: ugyanazt az ékezetes betűt több kód is megjelenítheti a képernyőn. Ilyenkor semmi sem mutatja, hogy valami különbség lehet közöttük, a nyomdai levilágító azonban mégis másként látja őket. A levilágítás során eltűnhet valami, ami a képernyőn még megvolt.

Azt gondolhatnánk, hogy ez csak szépséghiba. De mi történik olyankor, ha valaki úgy tudja, hogy elvégzett valamilyen javítást (hiszen a képernyőjén helyesen jelent meg), a szükséges változások mégsem mennek végbe? Mi van akkor, ha ez az adat inputként más folyamatokat irányít, vagy azoknak a működését befolyásolja?

Élő adatbázisoknál (amelyek például folyamatosan, a nap 24 órájában működnek) a kódoknak bármilyen változása súlyos következményekkel járhat. Nem véletlen, hogy számos helyen megmaradtak a CWI-kódnál, hiszen az adatbázis teljes átindexelése nemcsak időigényes, de rengeteg hibalehetőséggel is jár. Mekkorára dagadhatnak kicsinek látszó hibák is, amikor Windows alatt működő terminálok befolyásolják IBM nagygépek működését?

A Win95 hosszú fájlneveinek kezelése is járhat enyhén szólva kellemetlen következményekkel. Mi történik például olyankor, ha Win95-ről hosszú fájlneves fájlokat floppyra kimásol valaki, majd elvégzi a javításokat, de egy olyan gépen, amely nem ismeri a hosszú fájlneveket, és a javított anyaggal tér vissza a Win95-re? Ahol a rövid és hosszú fájlnevek egymás mellett élnek, ott a rövideken végre fognak hajtódni a változtatások, de a hosszúakon nem. Tessék végiggondolni, mi minden történhet, ha elmarad a változtatások átvezetése a hosszú nevű fájlokon...

Egyelőre ennyi talán elég is lesz. Legalábbis ahhoz, hogy mindenki elgondolkozhasson rajta, vajon olyan kicsiségek-e a szoftverekben meggondolatlanul vagy következetlenül elvégzett módosítások — akár a honosítással kapcsolatban kerülnek be a rendszerbe, akár attól függetlenül —, hogy nyugodtan legyinthetünk: majd a következő változatban kijavítják?

Vargha Dénes

A magyar nyelv ékezetes betűinek helye az egyes kódtáblákban

	CWI-2	CWI-1	Ventura	MSZ	Win-2	Win-1
á	160	160	160	160	225	225
Á	143	143	199	181	193	193
é	130	130	130	130	233	233
É	144	144	144	144	201	201
í	161	161	161	161	237	237
Í	141	140	205	214	205	205
ó	162	162	162	162	243	243
Ó	149	149	209	224	211	211
ö	148	148	148	148	246	246
Ö	153	153	153	153	214	214
ő	147	147	219	139	244	245
Ő	167	167	221	138	212	213
ú	163	163	163	163	250	250
Ú	151	151	214	233	218	218
ü	129	129	129	129	252	252
Ü	154	154	154	154	220	220
ű	150	150	220	251	251	251
Ű	152	152	222	235	219	219

A soknyelvű Európa

Nehéz falat a gépi fordítás

A gépi (vagy géppel segített) fordításnak két nagy potenciális felhasználója lesz a közeljövőben. Az egyik az Európai Közösség, ahol rendszeresen hatalmas mennyiségű munkaanyagot, dokumentációt kell hozzáférhetővé tenni minél gyorsabban, nagyjából azonos időben minden részt vevő ország képviselőinek, lehetőleg a saját nyelvükön. A másik maga a számítástechnika, amely különböző anyanyelvű felhasználói számára szintén folyamatosan nagy mennyiségű dokumentációt kell, hogy rövid idő alatt lefordítson vagy naprakésszé tegyen.

Az Európai Közösség egyik legnehezebben megoldható problémája a soknyelvűség. 1200 teljes állású fordító szorgoskodik jelenleg a közösség fordítószolgálatának irodáiban Luxembourgban és Brüsszelben, és aligha kell attól félniük, hogy helyüket rövidesen a számítógépek foglalnák el. Állandóan folynak a kísérletek a gépesítés optimális technológiájának kikísérletezésére, de mindeddig csak mérsékelt sikerrel.

Gépi fordítási kísérletek

Az első nagy kísérlet azt vizsgálta, hogy az üzemszerű fordítórendszerként propagált Systran mennyiben tudna eleget tenni a közösség követelményeinek. Az évekig folyó kísérlet eredménye meglehetősen lesújtó volt. A „nyers" inputból készített fordítást utószerkesztéssel elfogadhatóvá lehetett ugyan tenni, de a hibák kiszűrése aránytalanul sok emberi szakértelmet és erőfeszítést kívánt, ráadásul viszonylag sokáig tartott, és a hibák többsége annyira blőd volt, hogy az utószerkesztők pszichikailag is nehezen bírták a javítás rabszolgamunkáját.

Ezután elkezdődött egy nagystílű kutatási projekt azzal a céllal, hogy kifejlesszenek egy soknyelvű saját fordítórendszert, a Eurotrát. A rendszer alapkoncepciója az volt, hogy minden nyelv sajátosságait építsék bele egy virtuális, csak a gépen belül létező nyelvbe, s minden ország kutatói maguk készítsék el az erre a nyelvre és erről a nyelvről fordító programrendszert. Sokévi munka után részeredmények ugyan születtek, ám a kísérleti rendszer annyira lassú volt, hogy gyakorlati kivitelezését végül feladták.

Olyan próbálkozások is voltak közben, hogy az eszperantó töltse be egy ilyen „közvetítő nyelv" funkcióját. Ennek a programnak a megvalósításában a hollandok jártak az élen, akik nem kevés energiát és pénzt öltek bele a kutatásokba — eredménytelenül. Bebizonyosodott, hogy ezek a fejlesztések sem váltották be a hozzájuk fűzött reményeket.

Terminológiai adatbázis

Sikeresebbnek bizonyult az egyesülőben lévő Európa másik programja, amely a soknyelvű, szerteágazó terminológia összegyűjtésére és számítógéppel való kezelésére irányult. A rendszer tervezése még a 60-as évekre nyúlik vissza, a komolyabb fejlesztés azonban csak a 70-es évek közepén indult el. Ekkor került az Eurodicautomnak nevezett terminológiai adatbázis az Európai Közösség irányítása alá. Együttműködést építettek ki a Világbank terminológiai hivatalával, a Francia Szabványügyi Hivatallal és sok más intézménnyel, vállalattal.

A kidolgozás különböző munkafolyamai során (gyűjtés, feldolgozás, használat) bőségesen ellátják a terminusokat a nyelvi ekvivalenseken kívül a használatra vonatkozó információkkal, mintapéldákkal, és ahol szükségesnek látszik, definíciókkal is. A terminusok tematikai besorolásához kidolgoztak a pittsburghi Carnegie Mellon egyetemmel végzett közös kutatás keretében egy kifinomult, háromkarakteres osztályozó rendszert. Bevezettek egy „megbízhatósági kódot" is annak jelölésére, hogy a különböző nyelveken megadott megfelelők milyen mértékben tekinthetők ekvivalensnek, mennyire egyértelműek, szabványosítva vannak-e stb.

Új felállásban

Az elmúlt évtizedekben változó intenzitással, de töretlenül folyik az Eurodicautom fejlesztése, bővítése, korszerűsítése. A modernizációs program keretében az Európai Közösság Terminológiai Szolgálata — amely jelenleg a világ legnagyobb ilyen profilú intézménye — előzetes megállapodást kötött a stuttgarti Trados céggel egy átfogó fordítássegítő rendszer kifejlesztésére.

A fejlesztés alapját a Trados MultiTerm néven ismert soknyelvű terminológiamenedzselő programcsomagja képezi, amelyet már évek óta sikerrel használ 400 felhasználó az Európai Parlamenttel kialakított együttműködés keretében.

Az Európai Parlamentnek ugyancsak van fordítószolgálata, Luxembourgban és Strasbourgban, sőt 12 nyelvű terminológiai adatbázisa is, az Euterpe. A Trados az EP felkérésére most dolgozik azon, hogy az Euterpe adatbázist a MultiTerm szótár formájára konvertálja, és szabadon hozzáférhetővé tegye a szabadúszó fordítóknak is.

A Trados más megbízást is kapott az Európai Közösség Fordító Szolgálatától. A szerződés értelmében a Trados korszerűsíti és a MultiTerm formátuma szerint átalakítja a Eurodicautom mamut-adatbázisát, hozzáilleszti az Európai Közösségben szabványosított szövegszerkesztőt, a WordPerfect for Windowst, és elkészíti az adatbázisnak egy rendkívül sűrűre tömörített CD-s változatát. A kísérleti CD-t a hírek szerint már el is juttatták tesztelésre az európai kormányintézményekhez. (Egyébként az Eurodicautom hagyományosan kereshető szöveges adatbázisként már évek óta elérhető a Telneten keresztül.)

Az Eurodicautom jelenlegi állapotában mintegy félmillió terminust tartalmaz, 11 nyelven, legfőbb erőssége a mezőgazdaság, a gyógyszergyártás, a jog, a politika és a kultúra. Most folynak a tárgyalások számos állami intézmény és ipari vállalat csatlakozásáról, amelyek ugyancsak részt kívánnak venni az adatbázis továbbfejlesztésében. Különösen nagy reményeket fűznek az Eurodicautom használatához a jogi szabályozás és a kormányszintű anyagok fordításának területén.

Vargha Dénes

Kávéházi Café

A Java és holdudvara

Mint tudjuk, tele van a sajtó az Internetről szóló cikkekkel, ezekben pedig — még a nem szakembereknek szántakban is — meglepően gyakran találkozhatunk a Java szóval; valami homályosat írnak egy Java nevű nyelvről, amely csodaszer, és köze van mindenhez. Mi is hát valójában a Java, mire való, és ami a legfontosabb, milyen eszközökkel lehet használatba venni... már amikor egyáltalán érdemes.

A Java egy programozási nyelv, amelyet a Sunnál fejlesztettek ki (ez meg is látszik rajta, de erről majd később), eredetileg mindenféle elektromos eszközök vezérlő nyelve lett volna, semmi köze nem volt az Internethez, de véletlenül nagyon passzol hozzá. A nyelv leginkább a C++-ra és a SmallTalkra hasonlít, tehát nagyon objektumorientált. Ezt szó szerint kell venni, más módszerrel tulajdonképpen nem is lehet benne programokat írni. Mivel nem utódja semminek, ezért nem kötötték az elődök, tehát világos logikájú, szintaxis szempontjából egységes (leginkább a C-re hasonlít), és viszonylag könnyen tanulható.

Mitől illik a Java annyira az Internethez? Mivel platformfüggetlennek szánták, ezért a fordító nem gépi kódot, hanem ún. „bytecode"-ot generál, ezt minden egyes platformon az ott aktuális értelmező futtatja egy virtuális Java gépen. Ezek a bytecode-os fájlok kicsik, még viszonylag sokat tudó program ilyen kódja sem nagyobb 10-20 Kbájtnál, tehát nagyon gyorsan lehet ezeket letölteni a hálózaton.

Képzeljük el, hogy valaki a web-lapját kicsit dinamikusabbá akarja tenni. Például fényújságszerű, lassan mozgó szöveget akar megjeleníteni. Ha ezt mint képet kell átküldeni az Interneten, akkor beleőszülünk a várakozásba, ráadásul minden változtatásnál újra lehet az összes fázisképet generálni. De mi van akkor — és itt érthetjük meg a Java igazi ötletét —, ha nem képet küldünk át a hálózaton, hanem egy olyan programot, amely mindenkinek a gépén fut, kicsi, relatíve gyors, és ott helyben generálja a megjelenítendő fényújságot. Egyáltalán, mi van, ha a dokumentumok nem pusztán saját tartalmukat hordozzák, hanem a megjelenítésüket végző programot is. Ilyenkor egy új fájlformátumhoz nem kell speciális nézőprogram, maga a fájl tartalmazza ezt. Mi van, ha egy adatbázis-kivonat nemcsak az adatot tartalmazza, hanem lekérdezőprogram is van benne, és mindenki saját érdeklődésének megfelelően kérdezhet. Ez mind megoldható a Javával, méghozzá elegánsan és könnyen.

Mire számítsunk?

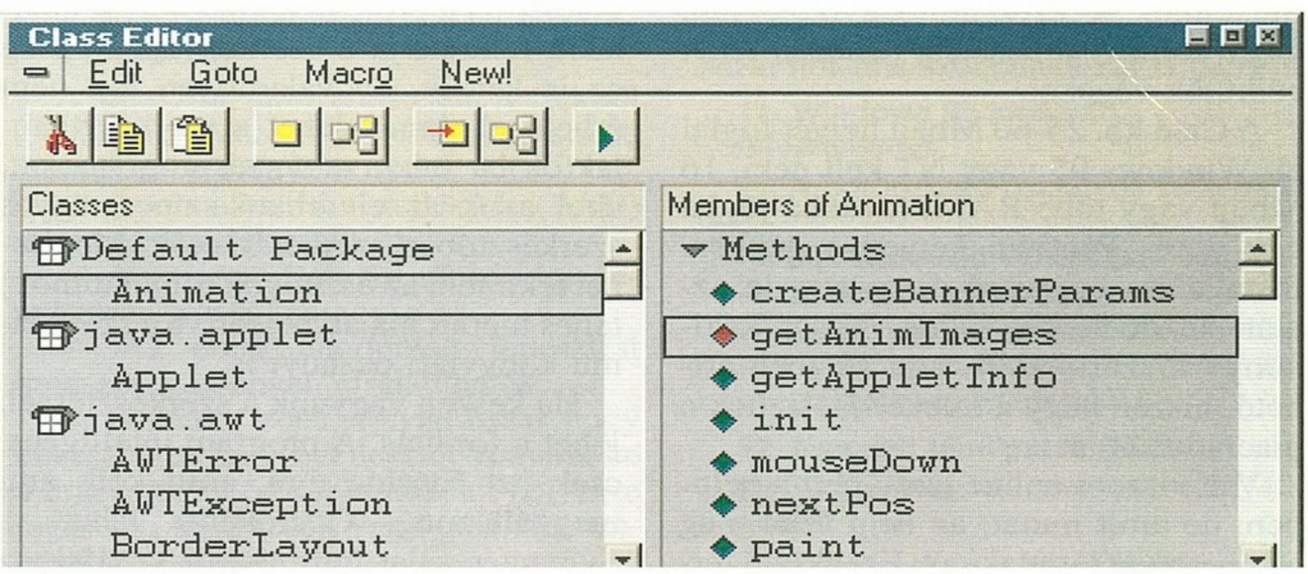
Persze rengeteg kérdés merül fel. Egyrészt, ha a Java interpretált nyelv, akkor lassú. Ez így van, de nem olyan nagy baj. Mivel ma már a gépek gyorsak, egy interpretált nyelv is elfogadható sebességet nyújt. Másrészt a legtöbb feladatnál, amelyet Javában oldanak meg, az idő nagyobb része azzal telik, hogy a program a felhasználó inputjára vár, és viszonylag nem nagy adatmennyiséggel dolgozik. Mi van, ha valaki egy rosszindulatú programot küld át a másik gépre, amely ott letörli az adatokat? A fejlesztők gondoltak erre: a Java a lokális gépen, ahol fut, nem tud hozzáférni a winchesterhez, és hálózaton is csak azzal a géppel tudja a kapcsolatot felvenni, amelyikről letöltötték. Továbbá a Javában nincs pointeraritmetika, sőt igazából pointerek sincsenek, azaz a legravaszabb programozó sem tud kinyúlni a védett virtuális gépből, hogy valami gonoszságot műveljen. Ettől a nyelv egyébként könnyebben kezelhető is lett. Még memóriafoglalás és elengedés sincs benne, a „szemétgyűjtést" a virtuális gép végzi. Ugyanez a virtuális gép ellenőrzi a letöltött bytecode fájlt, hogy nincsenek-e benne olyan hibalehetőségek, amelyek rosszindulatról vagy egyéb potenciális veszélyről tanúskodnak.

Nyilván így is ki lehet kerülni a védelmet, de ez mindennel így van, és a Java a körülményekhez képest nagyon biztonságosnak tűnik.

A nyelv mindezen kötöttségekkel együtt is nagyon erős. Noha ma a legtöbb alkalmazás csak tetszetős, de nem különösebben hasznos dolgokat művel, ne higgyük, hogy a Java csak játék! Lehet benne komoly programot írni, és írtak is már jó néhányat. Láttam már szakértői rendszert és online tőzsdei adatértelmezőt is.

Például az első javás képes böngészőt — tehát ami letölt és futtat Java-alkalmazásokat —, a HotJavát is Javában írták. Sőt a Sun teljes Java fejlesztő és nyomkövető rendszert írt Javában.

Ugyanis a programnyelv azt jelenti, hogy programozni kell. Milyen eszközök vannak a piacon, amelyek a Java-programozást segítik?



Na igen, a fejlesztés...

Megkérdezhetjük persze, hogy minek kellenek eszközök, ott az editor, a compiler, és rajta. A helyzet az, hogy a Java minden szépségével együtt sem könnyű nyelv. Nem is a szintaxissal van baj, bár abban is lehet hibázni, és akkor bosszantó a szerkesztés, a fordítás végtelen ciklusát éjeken át végezni. Egy szintaxis alapján színező editor sokat tud ilyenkor segíteni. Főleg grafikus környezetben futó programocskákat írnak Javában, tehát jó lenne valamennyire látni, amit éppen csinálunk. Továbbá az objektumorientáltság miatt nagyon bonyolult osztálystruktúrák vannak (még print utasítás sincs, az is valamelyik könyvtári osztály metódusa), tehát fejlesztés közben kellene látni (saját és a könyvtári) objektumstruktúránkat is.

A Java sikere miatt egyre-másra jelennek meg a piacon a fejlesztőeszközök. Mivel a nyelv új, és még változik is, ezért ezek az eszközök is sokfélék, bizonytalanok, változóak.

Meg kell jegyezni, hogy a java szó az amerikaiban a kávé egyik szinonimája (Java szigetéről sok kávébabot importáltak), ezért a legtöbb javás eszköz valami módon utal erre az eredetre. Grinder (daráló), Kawa, Beans (bab, babszem), Espresso, az amerikai névadás kimeríthetetlen találékonyságának termékei.

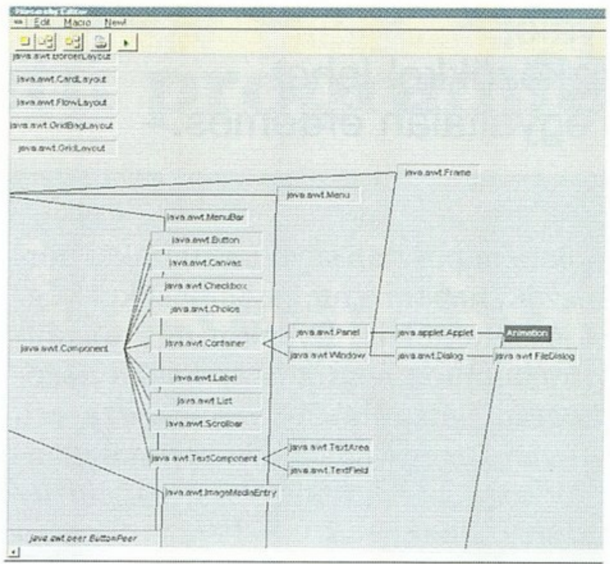
Az egyik legrobusztusabb kereskedelmi fejlesztőeszközt a Symantec jelentette meg Café néven. Mivel nekik volt már C++-os fejlesztőkörnyezetük, a két nyelv hasonlósága miatt nem volt olyan nehéz azt átültetni a Javára.

Café két verzióban

Van egy Lite, amelyet számos könyvhöz mellékelnek, és van a bolti változat, 150 USD körüli áron. A fő különbség a kettő között, hogy van-e debugger, vizuális fejlesztő és osztálybböngésző funkciójuk — ezek mindegyike olyan, hogy komoly fejlesztést nehezen lehet nélkülük végezni.

A Café kb. 25-60 Mbájt helyet foglal el, Windows 95 vagy NT kell neki, 16 Mbájt vagy több RAM, továbbá jó, ha egy gyors Pentium ketyeg a gépben. Installálása egyszerű és gyors, de azután kezdődnek a bajok. Nincs kézikönyv a szoftverhez, ami részben érthető, hiszen nagy a tülekedés, siettek a piacradobással.

Van viszont online tanítórész a helpben, de amit mutat, az nem felel meg a képernyőn látottaknak. Ezenkívül terjedelme kicsi, nem több, mint a főbb gombok értelmetlen magyarázata. (Nota bene gomb és ablak elég szép számmal van.) Ráadásul betöltöttem egy minta projektet, de az istennek sem akart futni. Lefordult, Cafén kívül mintha működött is volna, de azon belül nem, a hibaüzenet használhatatlan volt, így azután se nyomkövetés, se semmi. Hosszú próbálkozások után végül egy mellékelt readme-ből kiderült, hogy vagy rajta kell lennem az Interneten, vagy pedig egy obskurus Control Panel-beli TCP/IP-beállítást kell megadnom.



Miután ezt megtettem, a Café egyszer csak működni kezdett. Innen jöhetne az áradozás arról, hogy milyen remek is ez az eszköz. De nem ez jön.

Ugyanis e beállítások után nem tudtam felmenni az Internetre. Még sokat kísérleteztem, de a mai helyzet az, hogy vagy Internet, vagy Café. Talán erre is lesz megoldás egyszer.

Most már dicséret is...

A Café gyors, jól átgondolt szerkezetű, hihetetlen tömegű szolgáltatással. A fejlesztés minden aspektusa alaposan át van gondolva.

Munkáinkat projektekbe szervezzük, ezek tartalmazzák a forrásfájlokat (unixos ősök...), beolvasásnál a Café végigelemzi a projektet, valamelyik ablakába felrajzolja a szerkezetét. Itt még az is látszik, ha valamilyen fájl nem érhető el. Innen lehet a szerkesztőablak(ok)ba húzni a forrásprogramokat, ahol azok az elvárható színes, ügyes szerkesztőben szerkeszthetők. Meg lehet tekinteni az osztályok (objektumok) teljes hierarchiáját, beleértve a szépszámú könyvtári osztályt is.

Ha készen vagyunk a szerkesztéssel, jöhet a fordítás. A program intelligens, csak azt fordítja újra, amit kell, ami megváltozott. Opcióként lehet a Symantec saját fordítóját használni, ez kb. 5-10-szer gyorsabb, mint az eredeti Sun-féle. Rigorózus emberek maradhatnak az eredetinél. A lefordított program (ha átvergődtünk a TCP/IP-beállításokon, és lemondtunk az Internetről) jól futtatható és tesztelhető. A nyomkövetés során mindent lehet, amit egy intelligens debuggertől elvárhat az ember.

Van egy stúdió nevű rész, ahol nagyjából vizuálisan lehet összerakni a komponenseket, bár azért a Delphitől még messze van. Az egész rendszeren látszik, hogy a unixos múlt benne van. Windows95 ide vagy oda, mert persze az kell hozzá, a rendszer különféle környezeti változókra támaszkodik, ezért nemritkán kell újraindítani a gépet, hogy ezek a hatások érvényesüljenek.

A Caféval tulajdonképpen elfogadhatóan lehet fejleszteni, bár még nagyon érződik a Java újdonsága, valamint magának a szoftvernek a kiérleletlensége. Várom a javító verziót.

A többiek

Ne gondoljuk, hogy ma a Café az egyetlen a piacon, bár talán az egyetlen komoly kereskedelmi szintű termék. Forráskódszerkesztésre (és szerkesztőből futtatásra) ott van az ED, ez egyébként is figyelemre méltó, mindent tudó 32 bites editor. Még osztályhierarchiát is kukkolhatunk rajta.

A Microsoft készül — na, biztosan kitalálják a nevét — a Visual Javával, amelynek béta-változata most Jakarta álnéven fut.

Egy MOJO nevű eszköz — bár nincs annyi ablaka — szintén vetekszik a Caféval kényelmességben, csak kicsit sűrűn jönnek ki a 3-4 Mbájtos próbaváltozatok. Kísérleteztem a Sun saját Java Workshopjával, de legalább egy munkaállomás kellene hozzá, hogy elfogadható sebességet produkáljon, ráadásul percenként száll el.

A Borland nyomul saját C++-ának javás bővítésével. Kifejezetten a webes Java-alkalmazásokat célozza meg a (korábbi, San Jose-i cikkemben — Új Alaplap 96/6., 21. old. — említett) Jamba, valamint az Autodesk Hyperwire nevű fejlesztőeszköz. Még vagy 10 további eszközről hallottam (és naponta jönnek újak), de nem érdemes tovább sorolni.

A Java képes lehet átalakítani a mai számítástechnikát, és ha jó eszközök lesznek a fejlesztésre, akkor talán tényleg át is fogja alakítani. Addig érdemes megtanulni. Ehhez pedig egy editor és a compiler is elég lehet.

Horlai János

Tizenkét pont — fejlesztőknek

VisualAge for C++ for Windows

A VisualAge for C++ for Windows egy teljes C++ fejlesztési környezet, amely ötvözi a vizuális programozás nyújtotta kényelmet a professzionális fejlesztési lehetőségekkel. Az alábbi 12 pontban ennek részleteibe avatja be az IBM szoftveres termékmenedzsere a szakértő olvasókat.

1. Elérhetőség Windows platformokon is

Ezzel a nekünk talán furcsa nevű szoftverrel a fejlesztők tetszés szerint futtathatják C++ fejlesztési segédeszközeiket Windows 95-ön vagy NT-n, majd az elkészített alkalmazások Windows 95-ön, NT-n vagy Windows 3.1 alatt is futhatnak (Win32S API-kon keresztül). A program támogatást ad a Windows API-k számára, sőt OLE és ODBC is segíti a windowsos programozást.

Az IBM fordítója tehát — más segédeszközök kíséretében — megjelent a Windowson, így az IBM sokéves fejlesztési tapasztalatát a windowsos programozók is kamatoztathatják. Az OS/2-es fejlesztők számára pedig elérhetővé váltak a Windows platformok is, az OS/2-be fektetett munkájuk még inkább értékesíthető.

2. Összetett dokumentum-keretrendszer

Egyszerűbbé válik az OLE programozás. Egy keretrendszer többet jelent osztályok halmazánál, hiszen tartalmazza az OLE alkalmazások fejlesztéséhez szükséges kód nagy részét. Ezzel a keretrendszerrel az IBM az OLE alkalmazások készítésének nagyon egyszerű, de roppant hatásos módját adja a fejlesztők kezébe. Egyszerű, mivel egy OLE alkalmazás akár két-három kódsorral is elkészíthető. Hatásos mivel a teljes OLE specifikáció kihasználható.

A keretrendszer absztrakció megszabadít bennünket a közvetlen OLE API programozás keserveitől. A keresztplatform támogatás a portálást igen megkönnyíti. A rendszert úgy tervezték, hogy a fejlesztők könnyen áttérhessenek az OpenDoc technológiára.

Az IBM Open Class Library nagyon fejlett keretrendszer technológiát képvisel, mely az újrafelhasználás magasabb fokát nyújtja, mint más hasonló osztálykönyvtárak. A dokumentum-keretrendszer Taligent technológián alapszik. A Taligent technológiát több év alatt hozták létre, az IBM, az Apple és a Hewlett-Packard közreműködésével. Ezzel a Taligent keretrendszerrel lett kiegészítve azután az IBM Open Class Library. Még a legegyszerűbb OLE elem (control) hívások is megtalálhatók a keretrendszerben.

3. ODBC támogatás a Data Access Builderben

Egyszerűbb az adatbázisprogramozás, hiszen bármelyik relációs adatbáziskezelő ugyanúgy érhető el az Adathozzáférést Építő (Data Access Builder) segítségével, a Vizuális Építővel együtt használva pedig hatékony vizuális adatbázis-programozási eszközt kapunk.

A program mindegyik nagy adatbáziskezelőt támogatja, a DB2, a Sybase és az Oracle pedig az IBM által is le lett tesztelve. A fejlesztők további adatbázistámogatást is beépíthetnek alkalmazásaikba, melyekhez elérhető az ODBC meghajtó.

4. IBM Open Class Library

Objektumosztályok egyre növekvő halmaza segíti a fejlesztőket abban, hogy GUI, adatbázis, gyűjtemény és más alapvető alkalmazás logikai funkcióit egyszerűen újra felhasználhassák. Ezen objektumok újrafelhasználása egy nagyságrendnyivel megnöveli a programfejlesztés hatékonyságát. Több mint 400 osztály több mint 4000 metódusa található meg benne. (A portolhatóságot növeli, hogy az IBM Open Class Library sok más platformon is megtalálható.)

5. Vizuális építkezés a részekből

A fejlesztők előre elkészített részekből „rakhatják össze" az alkalmazásokat, aminek több előnye is van:

— Gyors alkalmazásfejlesztés (RAD) minimális kódolással.

— A programozók betanítási ideje lényegesen lecsökken.

— Az objektumorientált fejlesztési módszerek könnyen elsajátíthatók.

Ugyanakkor lehetőség van arra is, hogy kevesebb technikai fejlesztőt vonjunk be a fejlesztésbe, és a felszabadult helyekre jogászok, közgazdák vagy más szakemberek kerülhetnek.

6. IBM fordító-optimalizáció

Igen nagy a teljesítménynövekedés, a program 35%-kal gyorsabb, mint a legjobb versenytársaké. Windowsra optimalizált kód, a hardver optimalizációja i386-tól a Pentiumon keresztül a Pentium Próig. (Nagyon jó minőségű generált kód.)

7. Kimagasló keresztplatform fejlesztés

A VisualAge for C++ család más tagjai segítségével gyakorlatilag portolás nélkül, újrafordítással áttehetők az alkalmazások OS/2, AIX, Solaris, OS/400 és MVS környezetbe. A jelenlegi windowsos fejlesztés később továbbvihető más platformokra is, hiszen a VisualAge platformtámogatottsága tovább nő.

8. SOM Objects Development Toolkit

Az objektumorientált fejlesztéshez szükséges minden eszköz egy csomagban érhető el, amely új Windows NT támogatást ad egy harmadik verziónál tartó, kiforrott technológiával. (Itt is megvan a keresztplatformos objektumegyüttműködés az OS/2, Windows 3.1, Win95, NT, OS/400 és MVS platformokon.)

9. Alkalmazási „SmartGuide"

Az un. „SmartGuide"-ok olyan eszközök, melyek gyakran ismétlődő rutinfeladatok automatikus elvégzését segítik. A Project Sablongenerátor (Project Template Generator) a WorkFrame projektek elkészítését automatizálja. A Gyűjtemény Osztálygenerátor (Collection Class Generator) pedig az IBM Open Class Libraryban található gyűjteményosztályokkal történő kódírást egyszerűsíti le. (Hamarosan tervezési mintákkal is kiegészül, s azokhoz az Interneten keresztül lehet majd hozzáférni.)

10. IBM technikai támogatás

Az ügyfelek által nagyon jónak tartott technikai támogatás, mely elektronikus fórumokon és Interneten kívül ingyenes hazai hot-line szolgáltatással is párosul.

11. Windows 95 kinézetű felületelemek

— A hordozható osztályok támogatottak.

— Dinamikusan választható a kinézet.

— Erőforrás eszközök.

— Könnyen elkészíthető natív felület.

— OS/2-s erőforrások Windowsra konvertálásához segédeszköz.

12. Nemzeti támogatás (NLS)

Helyi nyelvek támogatása XPG4-en keresztül.

Nyikes Tamás

Sötét szobában fekete macskát kergetni?

Fájlkeresés az Interneten

Fpware:

FPARCHIE

Operációs rendszer: Win95
Kategória: Archie kliens
Dátum: 1996. július
Verzió: 0.8 béta
Méret: 240 K
URL: FPWARE

Néha előfordul, hogy az ember pontosan tudja, melyik fájlt keresi az Interneten (megvan a neve vagy legalább annak egy része), csak azt nem tudja, hogy az a fájl hol van, melyik szerver melyik alkönyvtárának mélyén rejtőzik. Ilyen keresésre találták ki az archie szervereket. Ezek rendszeres időközönként indexelik az FTP szerverek tartalmát, és egy speciális felületen keresztül, rejtélyes parancsokkal lekérdezhetők. Tehát jó tudni a keresett fájl nevének legalább egy részét, de ezen túl ismerni kell az archie szervereket és a parancsnyelvet is. Szükségünk van továbbá egy FTP programra, amellyel végül letöltjük a fájlt és mellesleg az FTP parancsait is.

A windowsos archie kliensek száma elég nagy, de az FPARCHIE messze kiemelkedik közülük. Felülete olyan, mint a Win95 Find Files programocskájáé, ami teljesen indokolt, hiszen végül is majdnem mindegy, hogy amit keresünk, az a mi gépünkön, a lokális hálózaton vagy az Interneten van. Megadjuk a keresett fájl nevét, és ha „fent vagyunk" az Interneten, akkor az FPARCHIE megkeresi nekünk a fájlt.

Néven itt nemcsak a fájl pontos nevét kell érteni! Ezt a legtöbb esetben úgysem tudja az ember. Elég a név elejét megadni, továbbá használhatók ún. dzsóker karakterek is. A név panelnél be lehet állítani, hogy pontos névre történjen-e a keresés, vagy a megadott nevet először pontosnak tekintse a rendszer, és ha mégsem az, csak azután keressen tovább. Szűkíthetjük a keresést földrajzi helyek szerint, például csak az afrikai szervereken levő fájlok iránt érdeklődve.

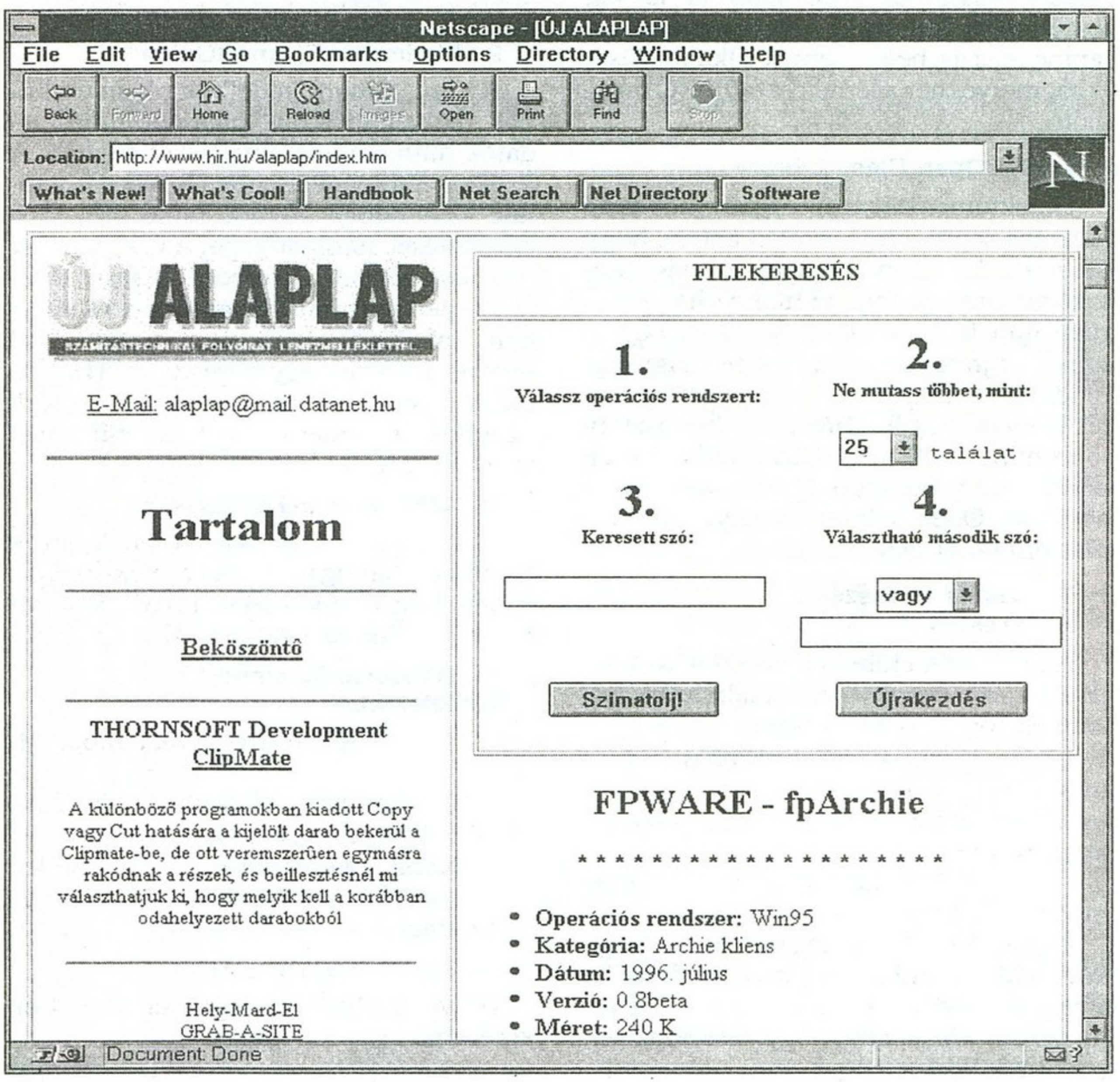
A hely nem tévesztendő össze az archie szerver helyével! A program előre konfiguráltan kb. 50 fontosabb archie szervert tartalmaz. A program egyetlen gyengéje, hogy egyszerre csak egy szerveren keres. Ha ott nincs hivatkozás (mondjuk a Rutgers szerverén) arra, amit keresünk, akkor sorra meg lehet próbálni a többit, például a funet.fi szervert. Szűkíthetjük a keresést a fájl dátumára, méretére is, és persze limitálhatjuk a találatok számát (ami alapértelmezésben 100). Így egy program friss verzióját könnyű kiszűrni az összes közül, hiszen kérhetjük, hogy csak három hónaposnál nem régebbi fájlokat jelezzen találatnak. Még arra is meg lehet kérni, hogy az alkönyvtár neve, amelyben a keresett fájl tartózkodik, tartalmazzon egy bizonyos karaktersorozatot.

Ha a jól ismert ablakban felsorakoztak a találatok, akkor végigböngészhetjük a neveket, helyeket, dátumokat stb., és a birtokunkban lévő adatok alapján el tudjuk dönteni, hogy végül melyik fájlt töltsük le. Itt jön a program következő kellemessége, ugyanis nem kell külön FTP programot elindítani, mert az be van építve az FPARCHIE-ba. A kívánt fájlra jobb gombbal kattintva kérhetjük rögtön a letöltést, és mintha az Internet és a mi gépünk egy volna, a jól ismert szárnyas animációval láthatjuk a letöltés folyamatát. Ezen a panelen követhető a sebesség és a várható letöltési idő is. Ha az FTP netán sikertelen volna, akkor erről értesít a program, sőt a részletes hibaüzenet is látható.

Keresgéléseinket megkönnyítheti, ha előzőleg belenézünk néhány shareware katalógusba (ott akár rögtön megtalálhatjuk, amit keresünk). Néhány ilyen a legismertebbek közül: CWSAPPS, TUCOWS, SHAREWARE.COM.

Figyelembe véve, hogy az FPARCHIE még csak a 0.8-as verziónál tart, ez a program az egyik legértékesebb segédeszköz lehet az Interneten böngészőknek. A végleges változat további újdonságokat is tartalmazhat.

Horlai János



Kritikai vitriológia, avagy hírek ellenfényben

Szoftverapróportéka

RAM-sokszorozók, avagy csak a cashemet figyelje, csalok: Hurricane 2.0 a Helix Software-től, Agent 95 a Hurricane-től és a MagnaRAM 2.0 feletti változata. Ha kézzel foghatóan is megjelennek, azonnal lerántjuk róluk a leplet. Mindenesetre e szoftverek bármelyikének ára nagyjából megegyezik 8 MB RAM-éval, és tudjuk, milyen jó érzés bűvészkedni 8 MB RAM adta keretek között is. Öreg, 16 bites Windows-zal.

❊

Az új IBM OS/2 WARP fedőneve: Merlin. Nyár legvégére ígérik. De lehet, hogy a nyár idén kihúzódik a Nagy Kék időszámítása szerint novemberig. Híreink szerint a Merlin nem lesz a Windows 95-tel kompatibilis. Akkor mi lesz? Néhány Merlinnel egybecsomagolt játékszer ínyenceknek?

▼

Kedit for Windows 1.5. Hosszú fájlnevek kezelése, OK. Web-szerkesztő HTML templating, ez úgy kellett, mint halottnak a beöntés.

☛

A web-témára mindenki rácsapott, mint gyöngytyúk a talonra. Az ausztrál Anawave Software HotDog Professional 2.0-ja nyomába eredt a sokkal profibb HoTMetaL Pro 3.0-nak, de a nyomukban liheg a Corel WEB.Graphics Suite, benne WEB.Designer, WEB.Gallery, WEB.Draw, WEB.Move, WEB.Transit, WEB.World, a sok WEB.adta. Ettől a sok WEB.tenyésztőtől bizony súlyos WEB.adót hajtatnék be. Mert WEB.ül szerzett hírnév WEB.ül vész el. WEB.adta, WEB.elvette. Egyik kutya, másik WEB. Na WEB.ből mostmár tényleg elég.

❀

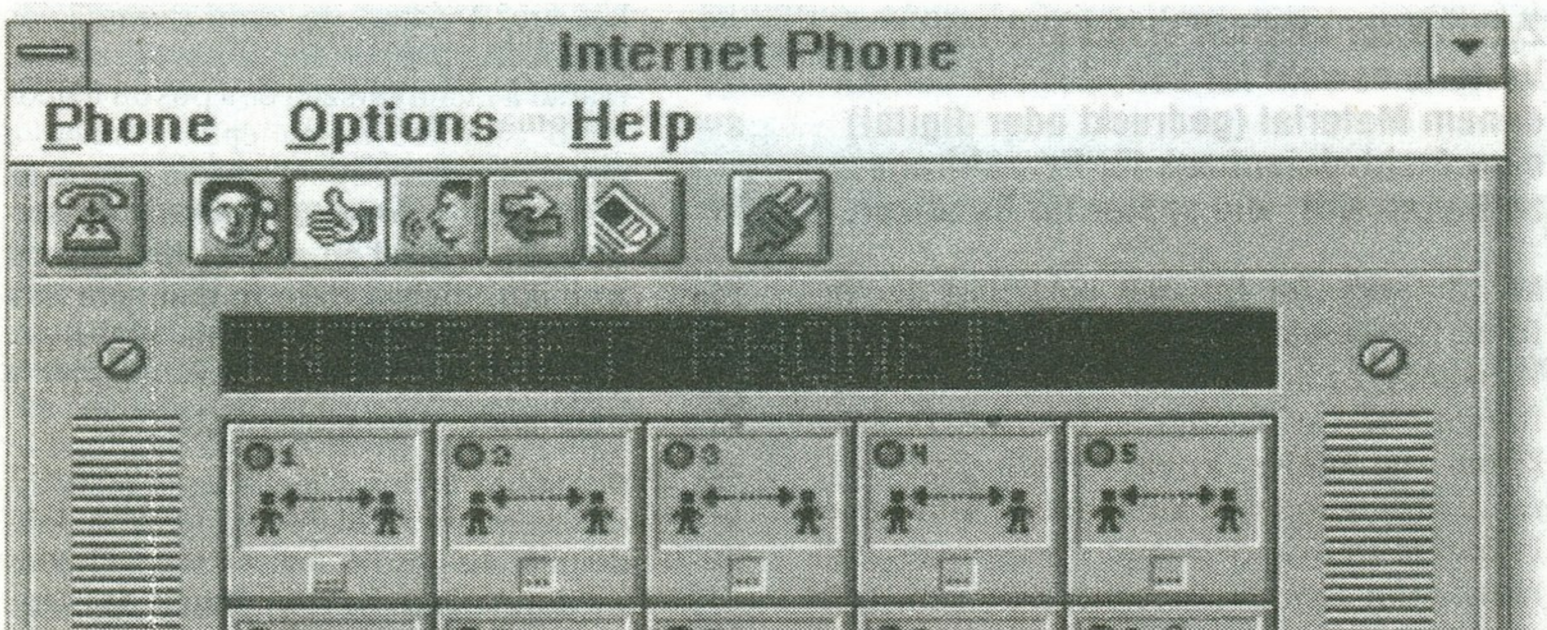
A Microsoft Bookshelf 1996-97-es kiadása már az Internetről táplálkozva hízlalható. Microsoft FrontPage: mire jó e LŐLAP? Ahhoz, hogy először lövünk egy nagyot, majd köré rajzoljuk a céltáblát. Így mindig bizton állíthatjuk, hogy a miénk telitalálat.

✆

WEB.ura fakó, van még jópár a kielégíthetetlen internetbuzik, szörcsögők (írd: search-ögők) és szörfösök oldalvizén evezgető cég és torzszülemény, eleve elvetélt ötlet, úgymint a Starfish Internet Utilities. QuickMarks, azaz kedvenc website-jaink könyvjelzőgyűjteménye, amelyek között már csak másodlagos könyvjelző segítségével tudunk majd eligazodni. NetMeter, szemünk előtt dagad a telefonszámlánk és ketyeg internetszolgáltatónk taxamétere. Kizárólag mazochistáknak vagy machopistáknak. NetClock. Pontosabb, mint az atomóra. Állítsuk óránkat a csillagokhoz, vagy ha azokhoz nem, hát a tengeri csillagokhoz. Philip Kahnnak, az ex-Borland atyaistennek pedig kívánom, hogy nyelje le a szaxofonját keresztben és on-line.

✂

Vegyük például a Mustang Software HTML-t tudó 5.0-ás WildCat BBS-ét, amely NT és Win95 fölött fut. Egy BBS-ben az a jó, hogy ha arra egy kuncsaft sikeresen rákapaszkodott, úgy kilépéséig vagy elejtéséig az összes erőforrás csakis az övé, nem szimultán megy a móka. Ügyes kis szeparé az Internet kávéházban. Az élmény és az íz azonos, de mindez olcsóbban.



✌

Bővül a Windows NT-n korrekt módon futó natív programok száma, ha nem is olyan nagyságrendben, mint az a Windows 95-nél gombamód történik. A

programok is a fényre jönnek (megjegyzem, a bugok is!). A Microsoft lassan átirányítja, áttereli a fényt az NT irányába. Egyre több rezidens vírusgát készül (McAfee VirusScan 2.5 for NT, F-Prot, IBM Antivirus stb.). Működő (nagy szó ez, valóban működő) hálózati modemmegosztó szoftver, a Sparta-Comm SAPS-a. Faxszoftverekre egyaránt jól és megbízhatóan funkcionál. Még az Internet is problémamentesen átcsordogál rajta. Valószínűleg a Microsoft is meg fogja venni. Bár elegánsabb lenne bírósági úton megtámadni, mint az NT kommunikációs szolgáltatásainak visszafejtése révén készült fejlesztést, és utána a pervesztes kishalat egy lesajnáló ajánlattal bekebelezni és utána kibelezni.

A korábbi kizárólagos Microstation disztribútor (a világ numero 1-es CAD programja, amely 4000 dollárnál kezdődik), az Intergraph 500 dolláros kezesbáránnyá szelídített kétdimenziós rajzoló- és tervezőprogramja, az Imagineer Technical (egyenesen Win95-re és NT-re) alaposan arcul csapja az Autodesk AutoCAD LT nevű, hasonló célú és árfekvésű programját. Sarcedzett felhasználók ezt is megvehetik.

Az Autodesket másik oldalról is érik erőteljes támadások. A 4000 dolláros 3D Studio Maxot balról előzi ár/teljesítményben az 1500 dolláros LightWave 3D 5.0 a Newtec cégtől. Animációkészítésben, gyors renderelésben a LightWave maga mögé utasítja vetélytársát, igaz 64 MB RAM alatt betölteni sem érdemes a programot.

A NetScape Navigatornak nem erőssége az e-mail kezelése. Hogy csak egy életből ellesett példát említsek, nem megy az, hogy elkezdek egy üzenetet írni, elmentem, később folytatom. Vagy elküldöm mentés helyett, vagy elvész. Nyögvenyelős az e-mail címkatalógus kezelése is, egy új név felvétele még mindig rébusz. Ilyen nyilvánvaló dolgok karikacsapásszerűen mennek az Eudora Pro for Windows 2.2-ban, amely a QualComm shareware-ként is elérhető gyártmánya.

Képfeldolgozás futószalagon? BMP, JPEG, GIF, TARGA, TIFF, CMYK, RGB és EPS állományok oda-vissza konvertálása (PC és Mac között is!), közben átméretezés, fényesség, élesség, kontraszt, színek állítása, mindezt a gép magára hagyva végzi a háttérben, míg az alkotó pihen (vagy on-line múlatja az időt a hálón). Ez a Gryphon Software Corporation „Batch It!" — nem túl fantáziadús — elnevezéssel illetett terméke.

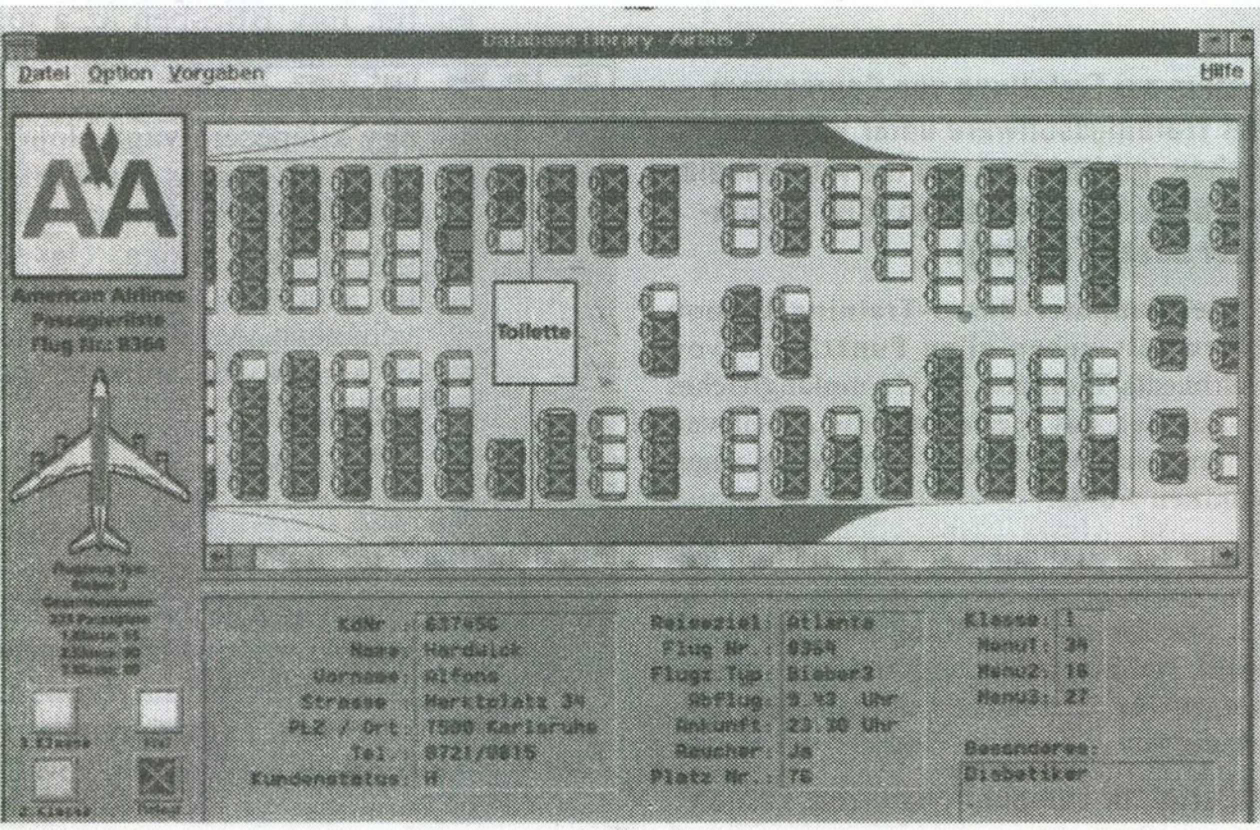
Norton NT Tools 1.0, szerintem a helyes verziószám 0.7 lenne. Mert akad a szerszámkészletben Norton File Manager (mint a Win95-ös Norton Navigatorban), antivírus szoftver, FTP keretprogram on-line fájlletöltéshez, erőforrás-kijelzőke, de kimaradtak a harddiszk- és hibajavító, valamint állománytöredezettséget megszüntető (defragmentáló) segédprogramok. Éppen ezért nem lett a csomag neve Norton Utilities for Windows NT 4.0.

Totally Twisted After Dark Screen Saver. A Berkeley Systems képernyőkímélői fogalommá váltak a számítástechnikában (különösen a repülő kenyérpirítók). Ezúttal arcpirító, sőt gyomorszorító dolgokra vetemedtek. A 13 új, teljesen animált, tetszés szerint paraméterezhető screen saver mindegyike egy ártatlan, már-már idilli képpel kezd. Bungee jumping. A gumikötélre erősítve magukat magasról a mélybe vető bátrak sportját űző tehénke ugrál, mígnem a gumikötél elszakad. Fröccsen a vér, törik a csont, horror a köbön. Vagy egy aranyos, tenyérbe mászó cicuska játszadozik a puha selymes fűben, és akkor egyszercsak jön a fűnyíró. A többit az olvasó kellően edzett idegrendszerére és képzelőerejére bízom.

Csattanós válasz a Maxis cég SimCity, SimCity 2000, SimLife, SimFarm, SimTrain, SimTower (magyarul SimOnTornya), SimAnt, SimTown, SimEarth stb. többszörös, ama bizonyos rókáról történő egynél többszöri bőr lehúzásának esetére: a LucasArts (lyukas arc?!) AfterLife című szimulációs játéka. Az ember sorsa bevégeztetett, marad a mennyország és a pokol. Mindkettő SimMetrikusan épül. Jönnek a rossz, elveszett, nyughatatlan lelkek, az Apokalipszis lovasai, a villany- és gázszámlás. Szóval pezseg az élet, így ezt a cinikus opuszt tekintve már nem az a kérdés, hogy van-e élet a halál után, hanem hogy egyáltalán van-e élet a halál előtt?

Végül az orvos (Dr. Windows) válaszol... Egy tisztelt felhasználó imígyen panaszkodott a minap: már megint nem tud az Excelből fekve nyomtatni. Válasz: próbálja meg egyszer talán ülve, majd innen kiindulva óvatosan hátra dőlve és excelcsak sikerül... Tényleg, mennyi is a fekve nyomás világcsúcsa?

Herczeg József



Betölteni egy piaci űrt

Integrált szerver, univerzális adatok

A közelmúltban bejelentett legújabb irodaautomatizálási rendszer az Oracle InterOffice szoftvercsomag. A költségtakarékos, könnyen kezelhető, kritikus alkalmazásoknak megfelelő rendszer az IBM AS/400 és a Microsoft BackOffice által (ki)hagyott teret foglalhatja el.

Az Oracle InterOffice csomag mind vállalati, mind munkacsoport-verzióban könnyen és gyorsan, a szervezet méretétől függetlenül teszi a webes és munkacsoportos alkalmazásokat a felhasználók asztalára. Az Oracle7 adatbázisra épülő csomag szorosan integrálja az adatokat, a különféle szöveges információkat, a képeket, a videót, a web-üzeneteket, a dokumentumkezelő és a workflow-technológiákat, a hálózatot, az adatbáziskezelőt, így biztosítva vállalati kategóriájú számítástechnikát egy igen könnyed „point-and-click" alkalmazási lehetőséggel.

Web-kapcsolat

Az Oracle szakértői szerint kis cégek és nagyvállalatok már egyaránt élvezhetik azokat az előnyöket, amelyek a meglévő szoftverek együttes működéséből, integrálásából származnak: „Az IBM már próbálkozott az AS/400-zal, de az a megoldás minden felhasználói helyen egy jól képzett információtechnológiai szakembert igényelt. A Microsoft BackOffice terméke pedig nem megfelelően integrált, és nem méretezhető".

Az új termékcsomag erőteljes, interaktív információkezelési és kommunikációs lehetőségei könnyen integrálják a legújabb és a jövőbeli web-alkalmazásokat a ma kliens/szerver rendszereivel és a tegnapról örökölt számítástechnikai rendszerekkel. Könnyed, központosított kezelhetősége szükségtelenné teszi az installációs teamek és helyi adminisztrátorok foglalkoztatását.

Ráépülve az Oracle7 adatbázis-technológiára, az Oracle InterOffice csomag univerzális adatválasztékot biztosít a felhasználóknak, beleértve szövegeket, dokumentumokat, képeket, audiót, videót és egyebeket.

Üzenetküldés, csoportmunka, új technológia

Az Oracle InterOffice csomag integrálja az Oracle korábban bejelentett, önálló termékekként megjelent, új üzenetküldési technológiáit. Az új csomag átfogó, méretezhető információkezelési platformot nyújt az egész vállalati szervezetnek, miközben megfelelő elérést biztosít az elektronikus levelezés, a naptárfunkciók, a táblázatkezelés stb. felé.

Támogatja a különféle kommunikációs eszközöket, a vállalati hálózatokhoz vagy a World Wide Webhez csatlakoztatott személyi számítógépektől, hordozható laptopoktól kezdve egészen a videotelefonokig és a pagerekig.

Az Oracle InterOffice készlet magába foglalja az Oracle WebSystem termékcsalád elemeit, és megfelelő hozzáférést biztosít az információkhoz a vállalati tűzfalak mindkét oldalán. Integrálva a csomag hatékony információkezelő rendszerét a Web lehetőségeivel, könnyű használatával, lehetővé válik a valós idejű vállalati adatok elérése, és a dinamikus web-oldalak folyamatos frissen tartása. Használva a Sun Java-appletjeit és a Microsoft OCX szabványokat támogató — az Oracle InterOffice csomag részét képező, beépített — Oracle PowerBrowsert, a felhasználók könnyen fejleszthetnek olyan innovatív internetes alkalmazásokat, amelyek tovább bővítik a felhasználók információelérési lehetőségeit.

Részek az egészért

Oracle InterOffice Manager — Igen könnyen használható, integrált információkezelő eszköz a vállalati szervezetek irányítására. A bővített lehetőségekkel bíró Oracle7 Enterprise Managerre alapozódik, megadván a módot a különböző adatbázisok, hálózatok, operációs rendszerek kezelésére, menedzselésére. Az Oracle InterOffice Manager lehetővé teszi a cégek számára, hogy elektronikus úton osszák szét és installálják a különböző szoftvereket, mind a klienseknek, mind a szervereknek. Ugyanakkor biztosan kezeli a cégek szoftverállományát és licencmegállapodásait.

Oracle Enterprise InterOffice (vállalati) csomag — A számítástechnikai iparág egyik leggyorsabb és legjobban méretezhető adatbázisára, az Oracle7 Enterprise termékre épül, és számos korszerű opcióval együtt megvásárolható. Ilyenek például a már meglévő rendszerekből való elérést biztosító átjárók (gateway-k), a szimmetrikus replikáció, továbbá a párhuzamos lekérdezés (query), és a párhuzamos szerver-opciók.

Oracle Workgroup InterOffice (munkacsoport) csomag — Biztonságos megoldást kínál a kisebb szervezetek számára. Fejlett, jól integrált, de még könnyen kezelhető számítástechnikai környezet. Az Oracle7 Workgroup Server technológiára épülő csomag hatékony, megbízható alkalmazáskészletet és megfelelő eszközöket biztosít a kis szervezeteknek ahhoz, hogy eredményesen tevékenykedjenek a mai elektronikus világban.

Közös stratégia

„Az Oracle InterOffice termék igazodik a Hewlett-Packard által meghirdetett speciális — a kezelhetőségi, az elérhetőségi és a teljesítményi szintekkel kapcsolatos — szállítási stratégiához. Az Oracle és a HP között sikeres, hosszú távú együttműködés van érvényben.

Az Oracle a HP első számú információtechnológiai szállító partnere a HP RISC-alapú számítógépeinek az esetében, és a mostani bejelentés alapján hasonló szerepet játszhat a HP díjnyertes, Intel-alapú hálózati szervereit illetően is" — ez a HP egyik általános igazgatójának véleménye.

Csányi György

Egy elfeledett módszerről

Értékelemzés az informatikában

Az értékelemzés éppen fél évszázada indult el világhódító útjára. De valami okból hazánkat és szakmánkat — mind ez ideig — elfelejtette meghódítani. Hajdan a szakemberek beszéltek az operációkutatásról, és tudták is, mi az értékelemzés. De akkor még alig volt olyan gép, amely 1 milliónál több műveletet végzett másodpercenként, a kritikus utak, a szűk keresztmetszetek, a gyenge pontok — és az optimumok — keresésének és vizsgálatának igénye mégis (vagy épen ezért?) természetes volt.

Lehet, hogy mára a számítógépek hatalmas teljesítőképessége szorította háttérbe a kényszerből vagy tudományos kíváncsiságból mindenben a legjobbra törekvő szemléletet. Ma már alig akad valaki, aki tudja, hogy mi az operációkutatás, mi az értékelemzés; a kritikus utak, a szűk keresztmetszetek és a gyenge pontok — és persze az optimumok — keresését senki sem igényli. Pedig, éppúgy mint hajdan, ma is vannak kritikus utak, szűk keresztmetszetek és gyenge pontok is.

A „gyenge pont" fogalma

A kritikus út, a szűk keresztmetszet és a gyenge pont egymáshoz közeli, de nem azonos fogalom. A velük való megismerkedéshez célszerű a legáltalánosabbat, a „gyenge pont" fogalmát választani.

A tudománytörténet 1946-ra teszi az értékelemzésnek nevezett gyakorlati optimalizálási „technika" fogantatását. Arra az időszakra, amikor nyilvánvalóvá vált, hogy a második világháborúban elképesztő módon megdrágult nyersanyagok ára ha később csökken is valamelyest, tartósan magas lesz. A nagy nyersanyag-felhasználó vállalatok, mint a General Electric, kénytelenek voltak komolyan foglalkozni az anyagköltségek csökkentésével. 1946-ban a General Electric két vezető embere, Erlicher és Winnie nemcsak igyekeztek valamit elérni a nyersanyagköltségek lefaragása terén (ezt akkor rajtuk kívül többen is megtették), hanem az általános eljárás kidolgozásának lehetőségét is felismerték.

1947-ben megbízást adtak Milesnak egy általános módszer kidolgozására. Miles és munkatársai, Sredenscheck, Fountain és Horn sikerrel oldották meg a feladatot, megalkotva az először „value analysis", majd „value engineering" néven elterjedt módszert, amelyet a hazai szaknyelv értékelemzésként ismer.

A módszert nagyon hamar általánosították, és nemcsak termékekre, hanem szolgáltatásokra is alkalmazták.

Az értékelemzés lényege, hogy a terméket, szolgáltatást két szempontból nézi. Az egyik szempont az, hogy mennyi értékhez jut a vásárló a termék vagy szolgáltatás által számára biztosított használati funkciók révén. A másik szempont pedig az, hogy mennyit fizet mindezért.

Az értékelemzés klasszikus formulája (tulajdonképp fő értékelőfüggvénye, fő célfüggvénye) az

```
érték = funkció(érték) /
ráfordítás(érték)
```

inkább általánosan értelmezendő, úgy, hogy a funkció(érték) és a ráfordítás(érték) viszonyát kell vizsgálni. Természetesen semmi akadálya nincs annak, hogy ha sikerül alkalmas módon számmal jellemezni a funkcióértéket és a ráfordításértéket, akkor a

```
funkció(érték) / ráfordítás(érték)
```

törtnek, vagy akár a

```
funkció(érték) - ráfordítás(érték)
```

különbségnek is értelme legyen.

Az értékelemzés során gondos munkával elemzik, hogy a termék, a szolgáltatás milyen funkciókat milyen áron elégít ki, és azt is feltárják, hogy mi lenne a (reálisan elvárható) ideális, mind a funkciók, mind pedig ezek ára tekintetében.

Az áru vagy szolgáltatás ún. „gyenge pontjai"-ról több szempontból is beszélhetünk. Gyenge pont lehet például egy termék egyik funkciója, amelynek megvalósítása túl sokba (túl sok pénzbe, túl sok időbe stb.) kerül. Gyenge pontnak számít az is, ha egy fontos funkció nincs kellő mértékben kifejlesztve. Többszörösen gyenge pont az olyan funkció, amely az előbbi fogyatékossága mellett még túl sokba is kerül.

A számítástechnikában, illetve az informatikában is lépten-nyomon termékekkel és szolgáltatásokkal van dolgunk. Ennek ellenére ezek egzakt vagy egzaktságra törekvő elbírálása valamiféle hallgatólagos közmegegyezés alapján tabu. Talán azért, hogy így mindenki elmondhasson saját termékeiről, szolgáltatásairól minden jót, a versenytársakéról pedig minden rosszat? Kétségtelen, hogy ma a számítástechnikai, informatikai működés „leggyengébb pontja" az egzakt elbírálás. Az informatika mindenről kínál információt, de az informatikai eszközök, szolgáltatások minőségi jellemzőivel kapcsolatban egzakt eligazítást nem ad.

Szolgáltatásaink értékelemzése

Az értékelemzés az Egyesült Államokban és Japánban, különösen az iparban, a 60-as évek elejére közmegbecsülésnek örvendő „nemzeti sporttá" vált. Több mint 30 éve hazánkban is megjelent Miles világhírű értékelemzési könyve. Ettől kezdve néhány kiváló szakember és egy csomó szélhámos vette gondozásába az értékelemzést. A kezdeti lelkesedés azonban Magyarországon hamar kifulladt, és az értékelemzés mára szinte teljesen visszaszo-

rult a hajdanvolt érdekességek közé — inkább tekinthető elfeledett módszernek, mintsem a gyakorlatban használt eszköznek. Ebbe viszont nem törődhetünk bele.

Hozzá kellene fognunk a számítástechnikai és informatikai termékek és szolgáltatások szisztematikus értékelemzésének és egzakt minőségi bírálatának megalapozásához. Mindenekelőtt magukat a funkciókat kellene összegyűjtenünk és rendszereznünk, amelyeket termékeink és szolgáltatásaink egyáltalán nyújthatnak. Az egzakt funkciókatalógus elkészítése után jöhet annak vizsgálata, hogy pl. egy szoftvertermék melyik funkciót milyen ráfordítással képes ellátni. A termékeket funkcióspektrumokkal és költségspektrumokkal kell jellemezni.

Ki tudja ma egzakt módon megbecsülni például, hogy a piacon levő operációs rendszereknek valóban mik is a „gyenge pontjai"? De nemcsak az operációs rendszerekről nincs megbízható funkció—ráfordítás elemzésen nyugvó minősítésünk, hanem a programozási nyelvekről és ezek fordítóprogramjairól sem. A rajzolóprogramokról, az adatbáziskezelőkről, a szövegszerkesztőkről ugyanezt mondhatjuk. Ma már — a funkcionális tisztázatlanság miatt — azt sem lehet igazán megmondani, hogy egyáltalán mit is kell szövegszerkesztő szoftverterméken érteni.

A funkcióleírások hiánya

Az eligazodás, az eligazító szolgáltatás nélkülözhetetlen a jó minőségű munkához. Ehhez javítanunk kell a jelenlegi helyzeten. Ki kell küszöbölnünk a gyenge pontokat, figyelnünk kell a kritikus utakra, a szűk keresztmetszetekre, és ahol fontos, az egyenszilárdság elvének érvényesítésére is, mert csak így közelíthetjük meg a számunkra elérhető optimumot. Minél tovább halogatjuk ezeket az elemzéseket, annál távolabb kerülünk attól, hogy az informatikai szolgáltatások arányosak legyenek az árral. Vagyis mindenért sokkal többet fogunk fizetni, mint amekkora értéket cserébe kapunk.

Remélhetően megnövekszik azonban a jelentősége az olyan funkció-összegző kísérleteknek, amilyen például Mohay Tamás cikke volt (A követelmények adatbázisa, Új Alaplap 96/7, 7. oldal). Más szakterületek funkcióleírásainak is bízvást ad helyet hasábjain a lap. Esetleg éppen ezek bázisán épülhet ki az az egzakt értékelemzési rendszer, amely ma még annyira hiányzik.

Pogány Csaba

Lapár és hirdetési tarifa

Leltár

Olvasóink megszokhatták már, hogy ebben a rovatban igyekszünk beavatni őket, a házunk táján történő, de mindenkire tartozó dolgokba, és azok hátterébe. Most úgy jött össze, hogy főleg a „piszkos anyagiakról" kell szólnunk, azokról készítettünk egy kis koraőszi leltárt.

Múlt havi számunk „médialógusa" megpróbálta érzékeltetni, hogy egy szakma és az arra épülő szakmai lapcsoport kapcsolatrendszere mennyire bonyolult. Bár az időzítéshez semmi közünk nem volt, tény azonban, hogy összeállításunk lapzártájakor (mintegy illusztrációként) megszűnt a Vogel Publishing kiadványa, a Heti Chip. Ennek okaira a Chip magazin következő száma ugyan nem tért ki, de az olvasókat a havi Chip irányába terelő szöveg a hirdetési piac beszűkülésére hivatkozott. (Feltételezzük, hogy emellett az ingyenes terjesztésről az értékesítésre történő átállás sem volt olyan sikeres, mint amilyennek lennie kellett volna.)

Az [Új] Alaplap kezdettől fogva két lábon áll. Nagyon hűséges, stabil olvasók „adják össze" bevételeinek közel felét, a másik felét pedig azok a hirdetők, akik tudják, hogy hirdetési célcsoportjuk szerkezete nagyon jól megközelíti az Új Alaplap olvasótáborának összetételét. (Lásd erről elemzésünket az 1996. áprilisi számban, illetve annak egy grafikusan feldolgozott részét, amely mostani lapunkban a tartalomjegyzékkel szemközt látható.)

Mit lehet tenni, ha az eladott lapmennyiség egy bizonyos határon túl nem növelhető (a mi lapunk értékesített példányszáma 5 éve folyamatosan 7-8000 között van), hirdetőinek köre pedig behatárolt, mert racionális médiaválasztás esetén e lap inkább „csak" a számítástechnika, az irodatechnika és a távközlés cégei számára jöhet számításba. Ilyen beszorított helyzetben kényszerű kiút: a lap árának és hirdetési tarifáinak a piac által is elfogadható emelése. (Vagy a lap megszüntetése, amit eddig még sikerült megúsznunk.)

A költségek növekedésének és a hirdetési piac ingadozásainak ellensúlyozására kénytelenek voltunk hirdetési tarifáinkat idén szeptembertől 22-25%-kal megemelni. „Lemaradásunkat" jól mutatja, hogy a számítástechnikai lapok családjában továbbra is az Új Alaplap hirdetési felülete kerül a legkisebb összegbe (egy oldal fekete-fehér hirdetés 120 ezer forint), a fajlagos és minőségi mutatókat is figyelembe véve pedig különösen jól állunk.

Közel fele ekkora arányban nő lapunk vásárlóinak kiadása: az Új Alaplap a hírlapárusoknál ez év októberétől 396 forint lesz, vagyis 40 forinttal kerül többe, de lapkategóriáján belül így is messze a legolcsóbb marad. Az extra CD-melléklettel megjelenő számok a többletköltség miatt ugyan drágábbak, de az előfizetők a CD-t ráadásként kapják, és őket természetesen a „normál" áremelkedés sem érinti.

Nekünk több okból is igen fontos az előfizetés. Elég ezek közül itt csak a legnyomósabb érvet említeni: az előfizetőknek történő postázás költsége 50-60 forint körüli összeg, míg árusítás esetén a lap árából (!!!) a terjesztő átlagosan 40 százalék (!) jutalékot kap. (Tessék számolni, mennyi marad a lap előállításának fedezésére!) 1996 őszén szervezendő kampányunkban igyekszünk is minél több olvasót meggyőzni, hogy az ő pénztárcájának szintén jót tesz, ha áttér a kényelmes, biztonságos (pl. munkahelyi) előfizetésre.

Nyáron elindítottunk egy új akciót: egyes régi Alaplapokért vadonatújakat adunk, a Mikroszámítógép Magazin teljes sorozatáért cserébe pedig az Új Alaplap egyéves előfizetését kínáljuk (lásd erről a 44. oldalt). A Mikroszámítógép Magazinból még további 2-3 garnitúrára „vagyunk vevők", az Alaplap megadott számainak becserélését pedig egyelőre korlátozás nélkül folytatjuk, sőt később esetleg más számokra is kiterjesztjük. Néhányan megkérdezték, hogy ez az „Alaplapcsere" mire jó. Örömmel mondhatjuk — mert egy lap rangját ez mégiscsak jelzi —, hogy ma is jönnek hozzánk olyanok, akik szeretnék lapunk korábbi évfolyamait megvenni. Ha tehát saját készleteinket ilyen csereakciókkal „komplettírozzuk", akkor ezt a számunkra nagyon hízelgő keresletet is ki tudjuk elégíteni.

Faklen Pál

Vezetőváltás a Compaq-nál

Az ízig-vérig novelles Drajkó László „átigazolásának" hírére — vagyis hogy szeptember 1-jétől a Compaq Magyarország ügyvezető igazgatójaként tevékenykedik tovább — biztosan sokan felkapták a fejüket. De hát végül is egészen jó szolgáltatásokra lehet képes egy Novell hálózat Compaq gépeken is...

Kiváló háló

Tapasztalatok szerint a számítástechnikai rendszerek meghibásodásának 80 százalékát a vezetékrendszerek okozzák, vagyis — és ez az X-Byte-féle filozófia — ha a hálózat kifogástalan, akkor a hibák 80 százalékát már sikerült is kiszűrni! A strukturált kábelezési rendszerek tervezésében és kivitelezésében annak idején első hazai cég ma már keményen küzd a konkurenciával. Nemcsak passzív, hanem aktív elemeket is szállítanak hálózataikhoz, így többek között foglalkoznak kábellel, rack-szekrénnyel és patch-panellel. 1993-ban — a magyarországi szolgáltatók közül egyedül — megkapták a Kiváló Áru emblémát. Most az ISO 9000 minősítés megszerzése következik, és 1996 második felében épített strukturált rendszereikre élettartam-garanciát adnak. Egy sajátos kábelezésfelügyeleti rendszer, a Patch-View a végponti számítógépek vigyázó eszköze, és ez adatlopás ellen is védi a hálózatot.

Ékezetes üzlet

A mostani hónap témája kapcsán érdeklődéssel figyeltük azokat a külföldi híreket is, amelyek a soknyelvű szövegszerkesztéshez és az Interneten nemcsak angol nyelven történő kommunikációhoz kapcsolódnak. Mindkettőben az Accent tette meg az utóbbi időben a legjelentősebb lépéseket. A cég nyár elején megállapodott a CompuServe-vel, hogy az „Internet with an Accent" szoftvercsomagjának részévé teszi a CompuServe Information Managert is, amely alapértelmezésben az Accent Mosaic browsert fogja megadni keresőprogramként. Nem sokkal később az IBM angliai vállalatával kötött szerződést, melynek alapján az Accent jutalékot kap az „Internet with an Accent" útján az IBM Global Network szolgáltatására bejelentkező új előfizetők után (akik ráadásul 10 óra ingyenes Internet-használathoz is hozzájutnak). Emellett az IBM jelentősebb vállalati ügyfelei közül 2 ezren megkapják az Accent internetes szoftvercsomagját. Hasonló együttműködés jött létre az Accent és egyes Internet-szolgáltatók között Ausztráliában, Franciaországban és Izraelben is. Szóval a világméretű, korrekt ékezetes kommunikáció ügyében már „válámi ván". És hátha ez lesz az igazi!

C(BS)A

Magyarországon talán még mindig a (CA-)Clipper a legnépszerűbb DOS-os adatbázis-fejlesztő eszköz. Ezt többek között kis hardverigényének köszönheti, illetve annak, hogy a PC-k megjelenésének időszakában fekete verziói „exponenciálisan" terjedtek. A hazai forgalmazó PC Szoftver becslése szerint jelenleg is mintegy tízszer annyian fejlesztenek Clipperben, mint amennyi a regisztrált felhasználók száma. A PC Szoftver Kft, amellett, hogy a BSA-ba belépve reméli — többek között a CA — érdekeinek hatékonyabb képviseletét, más eszközöket is latba vet a szoftverbitorlás visszaszorításának elősegítése érdekében. A legfőbb eszköz a most is tartó Clipper-amnesztiaakció, amelynek részleteiről a cég BBS-én található bővebb információ (214-6653). A CA-fejlesztőeszközök idei seregszemléjére („PC-Szoftver — CA Technikon") egyébként szeptember 26-án kerül sor.

Támad a tintafecskendő

Még meg sem száradt a tinta (ink) a fotóipar versenytársaként színre lépő HP inkjet-újdonság (Paletta rovatunkban közölt) anyagának korrektúralevonatán, amikor a HP bejelentette, hogy az idei nyarat (is) szorgos munkával töltvén szeptember 1-jével egy másik tintasugarasat is ránk ereszt: a DeskJet 1600CN színes hálózati nyomtatót, amely nagyobb sebességével, jobb minőségével és kisebb költségeivel a profi színes lézerprintereket akarja kiszorítani az irodákból. A HP sokat emlegetett új tintasugaras nyomtatási technológiája a jelek szerint frontális támadásba lendült. Lassan már csak az Epson FX 1000 átütő ereje tud ellenállni a (tinta)sugár-fegyvereknek.

Sybase—Netscape

Az adatbáziskezelőket gyártó Sybase terméklicenc-, viszonteladási és integrálási szerződést írt alá a Netscape webszervereire. A bejelentést követően másfél dollárnyit emelkedett a Sybase árfolyama a tőzsdén. A Sybase azt tervezi, hogy későbbi SQL szervereibe a Verity cég Montain View nevű szöveges állományböngésző termékeit is integrálja.

Már az Excel is...

Sokáig úgy tűnt, hogy a vírusok a szövegfájlokat és az alkalmazások speciális formátumait megkímélik. Az erre „szakosodott" Concept víruscsalád azonban rácáfolt erre, amikor a WinWord-öt vette célba, most pedig a McAfee cég hírül adta, hogy az áldozatok sorába bekerült egy újabb nagy windowsos alkalmazás, az Excel is. Az Excel mostani vírusai ugyancsak a makrókon keresztül támadnak. Viszonylag egyszerű azonban a fertőzöttség felismerése: ha a Tools/ Macro opciót választva „auto_open", illetve „check_files" bejegyzéssel találkozunk, bizonyos a Laroux névre keresztelt vírus jelenléte. A Laroux ellenszere e sorok megjelenésének idejére már elérhető és letölthető lesz, akár önállóan, akár a VirusScan részeként a McAfee web-oldalairól, illetve BBS-éről.

Pie Slice Party

A fenti címmel október 11-13 között országos számítógépes „házibulinak" ad otthont a budapesti Mechatronikai Szakközépiskola. A parti célja, hogy az ország minden területéről összegyűljenek a fiatalok (akár egyénileg, akár csapatban), és az általuk készített bemutató anyagokkal (kompozíciókkal, „compókkal", „stuffokkal") összemérjék tudásukat, kreativitásukat, ügyességüket. A szervezők a partira minden géptípus képviselőjét várják, de verseny csak PC-n, Amigán és C64-en lesz, a következő kategóriákban: játék, grafika, ray-trace, négycsatornás zene, multisávos zene, intrók, demók és wild compók. A rendezvényről, a jelentkezés módjáról részletes információ található lemezmellékletünkön.

Ágyból webezni Philips-szel

A Philips sem aludta „nyári álmát". Bejelentette a „webtévét", melyet a WebTV Networks technológiája alapján fejlesztettek ki. Adva van egy doboz (amely lényegében lecsupaszított PC, benne modemmel és bizonyos szoftverekkel), azt csatlakoztatni kell egyrészt a telefonvonalhoz, másrészt a tévékészülék videodugaszához, és kész. A weboldalak elég jól néznek ki, a görgetést és a tördelést úgy módosították, hogy a szövegek tévénézési távolságból is olvashatók legyenek. Az egészet külön erre a célra készített távirányítóval lehet vezérelni. A központi szerver (természetesen) Kaliforniában, Palo Altóban van. A 329 dollárba kerülő Philips Magnavox október 1-jén kerül az üzletekbe.

APRÓHIRDETÉSI MEGRENDELŐLAP

Kérem, hogy az Új Alaplap következő számának Mikrobazár rovatában közöljék az alábbi szövegű apróhirdetést:

....

....

....

....

(Maximális terjedelem: 300 betűhely)

Alaplap Posta

MEGRENDELÉS

Az Új Alaplap 1996/9. számában a 23. oldalon ismertetett szoftvert postai utánvétellel megrendelem.

Név: ....

(Céq:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Az árat a küldemény átvételekor a kézbesítési díjjal együtt kifizetem.

Dátum:

/aláírás/

Előfizetés az Új Alaplapra

Az 1996/.......... számtól kezdődően előfizetem az Új Alaplap c. havi számítástechnikai folyóiratot

...... példányban, ❑ 1 évre, ❑ 1/2 évre.

Az éves előfizetési díj 3564,– forint.

Az előfizetési díj kiegyenlítéséhez:

❑ **Számlát** kérek (banki átutalással fizetek).

❑ **Átutalási postautalványt** kérek.

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

- - - - - - - - - - - - - - - - - -

/aláírás/

INFORMÁCIÓKÉRÉS

Az Új Alaplap 1996. szeptemberi számának hirdetéseihez

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

Beküldhető: 1996. szeptember 30-ig

0901	0914	0927
0902	0915	0928
0903	0916	0929
0904	0917	0930
0905	0918	0931
0906	0919	0932
0907	0920	0933
0908	0921	0934
0909	0922	0935
0910	0923	0936
0911	0924	0937
0912	0925	0938
0913	0926	0939

FELADÓ — Feladáskor kérjük bérmentesíteni!

A) Egyéni érdeklődő:

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cég: ....

Ügyintéző: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Bélyeg helye

Új Alaplap szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Belföldön díjmentesen feladható

Új Alaplap szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Minden PC-hez kell egy jó alaplap!

És egy Új Alaplap!

Belföldön díjmentesen feladható

Új Alaplap szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

FELADÓ: — **Feladáskor kérjük bérmentesíteni!**

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon: ....

❑ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

❑ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát.
A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11701004-20171649 számlaszám.

Bélyeg helye

Új Alaplap szerkesztősége
I., Márvány u. 17.
Pf. 571
Budapest 1539

Az Új Alaplap lemezmellékletének tartalma **1996 / SZEPTEMBER**

INFORMÁCIÓKÉRÉS: 0914 ▲

INFORMÁCIÓKÉRÉS: 0921 ▲

INFORMÁCIÓKÉRÉS: 0910 ▲

Nyári Novell-újdonságok

A Novell megkezdte a Novell InnerWeb Publisher és a NetWare Web Server 2.5 kiadásának szállítását. Az InnerWeb Publisher teljes megoldást nyújt a belső dokumentumok közzétételére egy belső, vállalati web-adatbázisban. A szoftver legfontosabb része a NetWare Web Server 2.5, amely lehetővé teszi a felhasználónak, hogy NetWare szerverét használja web-adatbázisnak. A piacon kapható leggyorsabb web-szerver, egy NetWare Loadable Module (NLM) az egyetlen ilyen eszköz, amely megengedi a felhasználónak a tallózást a Novell Osztott Címtárban (NDS).

A Novell szintén bejelentette a GroupWise WebAccess intranet megoldás szállítását. A GroupWise WebAccess-szel a GroupWise-felhasználó Universal Mail Boxát bármikor, bárhonnan elérheti az Internetről. Ez az egyetlen eszköz a piacon, amellyel a felhasználó ugyanazon a kezelői felületen dolgozva képes elektronikus levelet, hangos üzenetet, távmásolatot, naptári bejegyzést, meghívást küldeni és kapni, feladatot egyeztetni az Interneten. A terméket a GroupWise 4.1-et használók már ez év júliusától, a GroupWise 5-öt alkalmazók ez év végétől használhatják.

FTP: következő generációs IP-alkalmazások

Az amerikai FTP Software a TCP/IP alkalmazásainak olyan új verziójával készül megjelenni, amilyenek az IP protokoll következő generációját támogató első kliens programcsomagok lesznek. Az OnNet32 2.0 32 bites Windows alkalmazói programmal, amely az IP Version 6-ot fogja támogatni, a vállalatok belső intranet hálózatokat állíthatnak fel és működtethetnek az IP V.6 bázisán. Mivel az intranet az IP V.4-es protokollján alapul, a vegyes, IPV.4 és IP V.6 környezetek kezelésére is felkészítették az OnNet 32-t. Az új FTP-termékbe biztonsági funkciókat (IFTF/IPSEC, WinSOCKS 4.0) is beépítettek.

3Com: LANplex és ONcore rendszerek

A 3Com a gerinchálózatoknál, illetve hálózati munkahelyeknél használható „high function", illetve „boundary switching" megoldások jelentősen kibővített palettájával jelentkezett, lehetőséget adva az osztott vagy útválasztós hálózatokról a kapcsolt hálózatokra és a virtuális LAN-okra történő átállásra. A termékek több mint 20 új kapcsolási megoldást ölelnek fel; ezek LANplex és ONcore platformokra készültek különféle LAN technológiákhoz (kapcsolt Ethernet, Fast Ethernet, FDDI és ATM). A LANplex nevű „high function" kapcsolócsalád nagy teljesítményű kapcsolóplatform, amely a gerinchálózatokban történő alkalmazásra optimalizált ASIC és RISC technológiákat használja.

Mind a LANplex 2500, mind a LANplex 6000 non-blocking tulajdonságot nyújt, valamint átfogó funkcionalitást, amelynek része többek között az intranet-útválasztás és az elosztott kapcsoláskezelés. A LANplex olyan vevőknek megoldás, akik át akarnak térni az útválasztón alapuló „collapsed backbone" környezetről a kapcsolóorientált topológiára úgy, hogy egyúttal az adatforgalom szükséges figyelése biztosított legyen. Az integrált ONcore rendszer egy sokfunkciós platform, amely nyújtja mind a „high function" (gerinchálózati kapcsolás), mind a „boundary switching" (munkahelyi kapcsolás) megoldást. A tíz új ONcore high function kapcsolómodul hatékony támogatást nyújt az FDDI, az Ethernet és a Fast Ethernet kapcsolás területén. Mindegyik az ASIC technológián alapul.

A hálózat magjának ez a megoldása a VLAN konfiguráció minden intelligens lehetőségét elérhetővé teszi, és egy rendszeren belül összesen 256 független VLAN-t tud kezelni. Ezenkívül átfogó adatfigyelési lehetőségeket nyújt, többek között priorizálást, protokollszűrést és teljes RMON-támogatást (9 csoport) a RAP, a Transcend Enterprise Manager és a Traffix Manager segítségével. Mindegyik kapcsolásmodul maximum 32 ezer egyedi hálózatcímet tud egymástól függetlenül kezelni, és a Multi-Gigabit buszcsatolóra csatlakozik.

UB Networks: növekvő Geo család

Az osztott hálózatmenedzsment megoldások úttörőjeként ismert amerikai UB Networks cég kiterjesztette Geo termékcsaládját a 16 portos GeoRim/Edt nevű Ethernet desktop hálózati kapcsolóval, és a 12 portos GeoRim/E Ethernet switch továbbfejlesztett változatával. Az új GeoRim/Edt könnyen installálható, nagy teljesítményű Ethernet switch, amely minden egyes portján 10 Mbit/s-os teljesítményt produkál. A kapcsoló maximum 193 500 keret/másodperc sebességgel működhet, minden egyes portján támogatja az SNMP-alapú hálózatfelügyeletet, és felszerelhető opcionálisan 100BASE-T Fast Ethernet eszközök gerinchálózathoz vagy szerverhez való kapcsolásához. A továbbfejlesztett GeoRim/E Ethernet switchet hajlékonyságra, nagy teljesítményre és hibatűrésre tervezték. Gateway-vel rendelkezik, ATM eszközhöz való kapcsolódáshoz opcionális 100BASE-T, FDDI vagy ATM összeköttetésekkel, amelyeket a portok újrakonfigurálása nélkül lehet installálni.

A Green River kulcsa az NDS

Másfél évvel a Novell NetWare 4.1 megjelenése után rövidesen piacra kerül a Green River fedőnevű új verzió. Szakemberek, akik megvizsgálták az új NetWare béta-verzióját, azt mondják, megérte várakozni. A Green River ugyanis éppen a Windows NT Achilles-sarkát jelentő globális címtárszolgáltatás hiányára játszik rá, azzal, hogy a Novell Directory Services (NDS) osztott címtár szolgáltatást tovább bővítette, fejlesztette. Az új NetWare-verzióban többek között megjelenik az NDS Manager grafikus eszközprogram, amely az adminisztrátor számára megengedi, hogy monitoron is felügyelje több NDS-fa replikációját és particionálását. A Green River NDS-ében megnövelt objektumszámmal találkozunk. Az NDS 2–16 millió közötti objektumot képes támogatni. Ezenkívül az NDS Systems Management Service-ben olyan változásokat hajtottak végre, amelyek abban segítik az adminisztrátort, hogy az NDS felhasználói profilokban található járulékos információt háttértárolón rögzítsék és (ha kell) visszaállítsák („backup&restore").

Kovács Attila

Az SGML szabvány III.

A DTD nyomvonalán

Az előző részben megismerhettük az SGML dokumentumok szerkezetének leírását. Most azok a dolgok kerülnek sorra, amelyek vagy nem láthatók, vagy éppen más szerepel helyettük. Első hallásra ezért haszontalanoknak tűnhetnek, ám olyanok is felhasználják, akik még nem is hallottak róla.

Talán még emlékszik az olvasó az előző részben boncolgatott példára, a verseskötetre. Ha a költő használná verseinek tárolására az SGML adta lehetőségeket, akkor biztosan örömmel venné, hogy minden vershez hozzákapcsolhatjuk a vers készültségi fokát, azaz egy vers lehet piszkozat, de lehet kész is, sőt esetleg valaki már át is nézhette — például helyesírási szempontból. Egy irodalomkutató pedig egy ilyen verseskötetben minden vershez (verstöredékez) hozzá szeretne kapcsolni egy azonosítót. E jelzőket, magyarul attribútumokat az SGML kezdőtagon belül adhatjuk meg:

```
<vers id=V123 statusz="piszkozat"> ... </vers>
```

Az már a feldolgozóprogram dolga, hogy az így megadott jelzők alapján mit tesz, esetleg kigyűjti-e a piszkozatokat (hogy a költő tovább dolgozhasson rajtuk), az azonosítók alapján keresztreferenciákat készít, vagy valami egészen mást csinál.

A jelzőket ugyancsak a DTD-n belül definiáljuk, egyrészt meg kell adni, hogy milyen elem jelzőiről van szó, meg kell adni a jelzők nevét, típusát, és végül azt, hogy mit jelent, ha nem említjük meg az elem használatakor ezt a jelzőt. Lássunk egy példát:

```
<!ATTLIST vers
        id      ID                                  #IMPLIED
        statusz (piszkozat | lektoralt | kiadott) piszkozat >
```

Mint a példából is látható, itt az ELEMENT szó helyett az ATTLIST használatos, ezt követi az elem(ek) neve, majd jelzőnként három adat. Ezek közül az első a jelző neve (értelemszerűen itt egyszerre két azonos név nem szerepelhet), a második lehet egy kulcsszó, avagy a használható lehetőségek listája, mint ahogy a státus lett definiálva. Ez utóbbi persze csak akkor használható, ha már előre tudjuk, hogy milyen lehetőségek közül választhatunk. (Sajnos ez nincs mindig így.) Ebben az esetben a feldolgozóprogram majd könnyedén megfigyelheti azt is, hogy adataink (jelzőink) helyesek-e.

Ha erre nincs lehetőség, mert például előre nem ismert számú eset lehetséges, akkor a következő kulcsszavakat használhatjuk: ID (akárcsak a példában) — az adott jelzőnek minden esetben egyedi értéket kell felvennie (magyarul nem lehet két azonos azonosítójú vers); CDATA — tetszőleges szöveg szerepelhet a jelzőben; IDREF — egy mutatónak kell itt szerepelnie, mely egy elemre utal; NMTOKEN — alfanumerikus sztring; NUTOKEN — számmal kezdődő alfanumerikus sztring; NUMBER — szám lehet a jelző értéke; NAME — maximum 8 betűből álló szó. A harmadik adat, amely megadja, hogy mi legyen, ha nem jelöltük a jelző értékét, vagy lehet a második helyen felsorolt lehetőségek egyike mint alapértelmezés (default), vagy lehet a következő kulcsszavak egyike: #REQUIRED — azaz a jelző értékét mindenképpen meg kell adni; #IMPLIED — nem fontos megadni; #CURRENT — legyen az, ami legutoljára is volt. Ez utóbbi esetben például elég lenne csupán az első vers státusát megadni (ha a definícióban ez szerepelne és nem a piszkozat), és a többi sorra az itt megadott értéket kapná meg.

Az öt oldalnál nagyobb művekben illik kereszthivatkozásokat használni, mint például a napilapokban is, hogy „Folytatás az 5. oldalon", vagy amikor egy könyv végén szereplő magyarázatokra utal a szó mögött álló apróbetűs számmal. Normális esetben ezeket a hivatozási adatokat a mű készítése közben nem kell tudni, hiszen könnyedén megváltozhatnak az oldalszámok, felcserélődhetnek a fejezetek és így tovább. Persze az sem jó megoldás, hogy csak a végleges mű elkészülte után írjuk bele ezeket az adatokat.

Az előbb verselemnek definiáltunk egy azonosítót, amely minden vers esetén más és más lesz. Ez szükséges ahhoz, hogy egyértelműen tudjunk a versekre utalni. Most készítsük el az utaló elemet:

```
<!ELEMENT utalvers - O EMPTY          >
<!ATTLIST utalvers    cel IDREF #REQUIRED >
```

A korábbi cikkben (mivel nem volt rá szükség) nem említettem meg az EMPTY tagot; ez jelzi, hogy példánkban az utalvers-nek nincs semmi tartalma. (Ezért felesleges a zárótag használata is.) Viszont van egy 'cél' elnevezésű jelzője, amit mindenképpen ki kell tenni, és ez fog utalni a versekre. A korábban felhasznált vers eleje ezzel így nézne ki:

```
<vers id=medve><cim>Egy csekélyértelmű medvebocs verselménye
<strofa>
<sor>Hétfőn, mikor a hőség rekkenő,
```

Majd a kötethez kiadott tanulmányok forráskódjában a következő szerepelne:

```
A. A. Milne filozofikus versében <utalvers cel=medve> a ...
```

Az már megint a feldolgozóprogramon múlik, hogy a tanulmányban itt, az utalás helyén majd a vers címét adja meg, esetleg azt a oldalszámot, ahol ez a vers található, vagy a vers első pár sorát. Ezzel nem érdemes nekünk foglalkozni (főleg akkor nem, ha ugyanazt az utalást különböző matematikai folyóiratok különböző módon szedik ki). Az SGML ellenőrző-elemző programok (parsers) általában csak azt vizsgálják meg, hogy létezik-e az az elem, amelyre utaltunk.

Most pedig: egészen más!

Az SGML nemcsak arra való, hogy a szövegek feldolgozásának megkönnyítésére telepakolja azokat mindenféle megjegyzéssel a szerkezetükről, hanem arra is, hogy egyszerű és hordozható módon megjelölhessük az aktuális szöveg bizonyos részét. Erre az ENTITY kulcsszó használatos. (A szótár ennek a szónak igen sok fordítását tartalmazza, de egyik sem felel meg nekünk, ezért vagy az angol kifejezést használom, vagy rövidítésként utalok rá.) Lássuk például a következő definíciót:

```
<!ENTITY SGML "Standard Generalized Markup Language">
```

Ettől a definíciótól a forrásszövegben szereplő SGML betűszó helyett nem fog a végső dokumentumban a kibontott alak szerepelni. Ha ki akarjuk használni ezt a rövidítést, akkor a rövid alak elé egy & jelet, míg mögé egy ;-t kell tennünk (ha nem szó közepén szerepel a rövidítés, akkor a pontosvessző akár el is hagyható).

Az élesebb szemű olvasók talán már a korábbi részekben észrevették, hogy olyan egyszerű szövegek leírásával baj van, hogy 'x < y + 1', mivel a kisebb jelet az SGML kezdőtagnak tekinti, és ezért végül hibát okoz. Kicsit bonyolultabb szövegekben már magyar ékezetes, görög, cirill karakterek is előfordulhatnak (a matematikai jelekről nem is szólva). Mivel az SGML-szövegek a hordozhatóság miatt csak ASCII karaktereket tartalmaznak, valahogy kódolni kell a sokféle karaktert. Erre az ENTITY lesz a megoldás.

A kisebb jelet az 'lt' írja le, így az előbbi formulát az 'x < y + 1'. Ekkor problémát okozhatna az & szerepeltetése a szövegben, de erre való az &. Ezzel a módszerrel éri el a HTML is a magyar és egyéb karakterek ábrázolását. Nem csupán karaktersorozatot rövidíthetünk le ilyen módon, hanem egy ilyen szavacskával akár egész fájlra is hivatkozhatunk. Ekkor, hogy ne keverhessünk össze egy ilyen rövidítést a korábbival, a rövidítés és a fájlnév közé tegyük ki a SYSTEM szót. Ekkor a rövidítést kibontva, a fájl teljes szövegét kapjuk vissza.

Ha olyan a feldolgozórendszer, akkor a SYSTEM után található szöveget nem csupán fájlnévnek tekintheti, hanem például adatbázis-lekérdezésnek, vagy akár rendszerhívásnak, és ekkor a választ illeszti be a szövegbe. A lehetőségek szinte korlátlanok! Természetesen különböző gépeken ugyanazon név alatt más szöveg is szerepelhet, tehát egy forrásból helyszíntől függően más és más eredmény nyerhető (főleg az adatbázisoknál)!

Egy programhoz illik megfelelő dokumentációt is írni. Általában ez egy fájlban (könyvben) szerepel, és ebben minden megtalálható. Az egyszerű (vagy kezdő) felhasználók számára ennek csak egy része érdekes. Ezért érdemes ugyanazt több példányban is elkészíteni külön-külön minden szintű felhasználónak. (Ezzel a felhasználó is jól jár, és ha idővel megveszi a bővebb verziót, akkor a szerző is.) Mindez nem jelenti azt, hogy mindent többször is le kellene írni. Ebben az esetben a jelölt bekezdések segítenek. A következő kulcsszavakat használhatjuk:

INCLUDE — a bekezdés bekerül a végeredménybe;

IGNORE — ezt a bekezdést kihagyjuk;

CDATA — az itt található szöveg betű szerint kerül bele a dokumentumba;

RCDATA — az előzőtől csak annyiban tér el, hogy a korábban ismertetett rövidítéseket kifejti;

TEMP — ez a piszkozatban szerepelhet, csupán azt jelzi a szerzőnek, hogy ezt majd még át kell írni valamely korábbira.

Az is megoldás lehetne, hogy kézzel írjuk át a megfelelő INCLUDE szavakat IGNORE-ra vagy fordítva, ám felesleges, mert az

```
<![ INCLUDE [
A sorokat az ENTER billentyű leütésével zárjuk le.
 ]
```

helyett a dokumentációban

```
<![ %Kezdo; [
A sorokat az ENTER billentyű leütésével zárjuk le.
 ]
```

is szerepelhet <!ENTITY % Kezdo 'INCLUDE'> definíció után, és ezzel minden alapszintű számítógép-ismeret bekerül a dokumentumba. Viszont ezek teljesen feleslegesek a nagymenők számára, ezért a nekik írt változatban az <!ENTITY % Kezdo 'IGNORE'> fog szerepelni. Ezek a definíciók nagyon hasonlítanak a korábbiakhoz, csupán itt az és jele helyett a százalékjel szerepel kibontáskor, illetve definícióban is szerepeltetni kell a százalékjelet.

Mivel az egyszerű felhasználó számára más lényeges különbség nincs, ezért ezt nem is részletezem tovább, akit jobban érdekel, az nézze meg az előző lemezmellékletek DTD-példáit.

Részekből az egész

Eddig tartottak volna a részletek egy-két apróságtól eltekintve (amelyek fölött el fogok siklani a továbbiakban is). Ezek után már csak össze kell rakni ezekből az egészet.

Egy SGML dokumentum elvileg két részből áll. Az elsőben mindenféle dokumentáció szerepel, míg a másodikban maga a szöveg. Az első részben gyakran külön fájlok vannak (ezek kapják a DTD kiterjesztést), vagy például HTML esetén be vannak drótozva Netscape-szerű programokba. Mivel egy DTD megtervezése nem egyszerű feladat, gyakran már kész DTD-t szokás felhasználni. Belepiszkálhatnánk magába a fájlba is, de ekkor nehéz a megváltoztatott DTD-t szinkronban tartani az eredetivel (főleg, ha az gyakran megváltozik). Ezért inkább az a megoldás terjedt el, hogy újradefiniáljuk az eredeti DTD bizonyos részeit. Az eredeti szerkezeti elemeket (elements) nem változtathatjuk meg ilyen módon, de a rövidítések újradefiniálása bevett, mindennapi szokás.

A sima szövegben — amelyet majd az egyszerű felhasználó fog elkészíteni — már nem szerepel semmi definíció, ott csupán alkalmaznia kell a korábban definiáltakat. Egy terjedelmesebb dokumentumot nem érdemes egy fájlban tartani, az egyszerűbb kezelhetőség érdekében hasznos a fejezeteket (vagy tagolható egységeket) külön fájlokban tartani. Ekkor lesz egy fájl, amelyben egyrészt (rövidítésként) definiáljuk, hogy melyik fejezet melyik fájlban található, majd szépen felsoroljuk a rövidítéseket:

```
<!DOCTYPE konyv [
        <!ENTITY fejezet1 system "fej1.txt">
        <!ENTITY fejezet2 system "fej2.txt">
        <!ENTITY fejezet3 "-- ezt Béla még nem írta meg --">
]>
<konyv>
<elolap> ... </elolap>
<szoveg>
<bevezetes> ... </bevezetes>
```

```
<fejezetek>
    &fejezet1;
    &fejezet2;
    &fejezet3;
  </fejezetek>
</szoveg>
</konyv>
```

Most jön a feketeleves...

Ezek az ismeretek szerintem elegendőek arra, hogy egy tetszőleges DTD-t megértsen az olvasó, és annak megfelelő SGML fájlokat készítsen most már önállóan is. A vállalkozóbb kedvűek megpróbálkozhatnak saját DTD-k írásával is. Ha megnézünk egy tetszőleges DTD-t, akkor látjuk, hogy nagyon sok az ismétlődő részlet. Hiába készítjük fel kedvenc szövegszerkesztőnket ide illő makrókkal, még így is nagyon kényelmetlen elkészíteni a dokumentumtípus-leírást. Alapos felkészüléssel talán elsőre sikerülne megírni a végleges változatot, ám az emberek legalább 95 százaléka inkább az addig_próbálkozz_amíg_össze_nem_jön módszert használja. Ekkor pedig a keservesen megírt leírást majdnem teljesen újra kell írni.

A korábbi lemezmellékleteken közreadott EasyDTD megpróbál megszabadítani e folyamat unalmas, ismétlődő részeitől. Ehhez a dokumentumtípus-leírásnak egy rövidebb formáját használja N. E. Smith (akit az Sviewer, az Sengine, illetve az Until alkotójaként ismerhettünk meg). A rövidebb fogalmazás bizonyos korlátozásokat is hoz magával: a DTD-n belül az <ENTITIES>, <ELEMENTS>, <ATTRIBUTES> kulcsszó szerepelhet. Ezek után egy csoportban definiáljuk a megfelelő adatokat (ettől eltérően a DTD lehetőséget ad arra is, hogy egy elem definíciói után egyből definiálhassuk attribútumait). Az ENTITIES után a rövidítések következnek, az első oszlopban (azaz új sorban) kell kezdődnie a névnek, amelyet idézőjelek között követ a teljes, kifejtett alak. (Ez utóbbi akár többsoros is lehet.)

A DTD definícióban használhatóak azok a rövidítések, melyeket százalékjellel kezdünk. Az <ELEMENTS> vezeti be az elemek definícióját. Itt a elemek hierarchiáját a sor elején található szóközök jelzik (két szóköz jelent egy szintugrást). Az azonos szinten található elemek az előző szintű elem felbontását jelzik. Használhatjuk a zárójeleket (csoportosításra), illetve a már korábban bemutatott speciális jeleket. Vigyázzunk arra, hogy szóközöket az elemek neve és e jelek között ne tegyünk ki! A jelek használatát oldalakon keresztül lehetne részletezni, ám szerintem úgy lehet azokat legjobban megérteni és megtanulni, ha a program mellé adott példafájlok alapján elkészítünk egy-két sajátot.

Az <ATTRIBUTES> vezeti be a jelzők definícióit. Legegyszerűbb módon a sor kezdetére kiírjuk az elem nevét, majd felsoroljuk a jelzők neveit. (Ekkor a jelzők CDATA típusúak lesznek, és nem fontos szerepeltetni őket.) Ha olyan sok jelző lenne, hogy nem férnének ki egy sorba, akkor a következő sorban is folytathatjuk, csak e sor első karakterhelyét hagyjuk üresen. Ha más típusra is vágyunk, akkor soronként csak egy jelzőt írjunk az elem neve után, és utána soroljuk fel annak típusát és a hiányzó elem kezelését.

Miután elkészítettünk egy ilyen fájlt, és illendőségből EZY kiterjesztéssel láttuk el, már el is készíthetjük a DTD fájlt. Ez is remek dolog (elvégre ezért jött létre ez a program), ám ennél jóval többet is tud. Ha az –A opcióval futtatjuk, akkor elkészíti a Sengine számára ehető programot, amivel egyből le is lehet tesztelni szövegállományainkat, hogy tényleg illeszkedik-e rájuk a DTD. Az –Y opció segítségével egy olyan AWK programot kapunk, amely kibontja rövidítéseinket. (Ez csupán két érdekes opció volt a tizenkilencből.) Talán ennyi is elég annak bemutatására, hogy nem szükségesek dollártízezrek az SGML használatához.

Aszalós László

Kalóztérképektől az árnyéksémáig

Titokmegosztás

A titokmegosztási sémák hatékony segítséget nyújtanak informatikai rendszerek kritikus pontjainak védelméhez. A kritikus információ (kulcs, jelszó) szétosztása csökkenti a rendszer függőségét az üzemeltető személyektől, illetve véletlen adatvesztés esetén is védelmet nyújthat. Cikkünkben áttekintjük a klasszikus titokmegosztási technikákat, s megemlítjük az e területen kidolgozott újabb sémákat is.

Megosztott titkokról sokféle mendemonda kering. A Coca-Cola receptjének tudorairól például sokáig azt sem lehetett tudni, hányan is vannak (azt viszont igen, hogy egyszerre sohasem ülhetnek repülőre).

Titokmegosztási sémákat már a középkorban is használtak. Jó példa rá, amikor három kalózvezér elássa a rablott kincset, majd a kincs környezetéről egy térképet készít. A térképet háromfelé vágják, s mindenki eltesz egy darabot. Utódaik öröklik a térképdarabokat, és ha ők meg akarják keresni a kincset, a következő problémákkal kerülhetnek szembe:

Nem találják meg egymást, vagy valamelyik térképdarab elvész. Kétséges, hogy a maradék darabokból a kincs helye meghatározható-e, hiszen a három kalóz éppen azért darabolta fel a térképet, hogy egymásra legyenek utalva.

Ugyanakkor valószínűleg mindegyik térképdarab hordoz információt a kincs helyéről, hiszen ha nem így van, akkor az adott darab felesleges. Elképzelhető az is, hogy egyetlen térképdarab is ad annyi információt, ami lehetővé teszi az elindulást, vagyis a fennmaradó lehetőségek kipróbálását, végigjárását idő vagy pénz befektetésével. Ismert irodalmi példa az, amikor a tengerből kifogott palackposta a hajótörött szigetének csak a földrajzi szélességét közli, a hosszúsági fok olvashatatlan (J. Verne: Grant kapitány gyermekei). A hiányzó adat ekkor a Föld körbehajózásával pótolható, bár elég sok múlik azon, hogy az adott szélességi kör az Egyenlítőhöz vagy a sarkokhoz van-e közel.

Látható tehát, hogy a fenti módszerrel két probléma van. Egy szereplő kiesése, egy titokdarab elvesztése meghiúsíthatja a titokdarab rekonstrukcióját. A másik probléma az, hogy egy vagy két titokdarab már annyi információt adhat, hogy a fennmaradó lehetőségek kipróbálhatók. (A titok értéke dönti el, hogy megéri-e kipróbálni.)

Hogyan csinálnánk ezt ma?

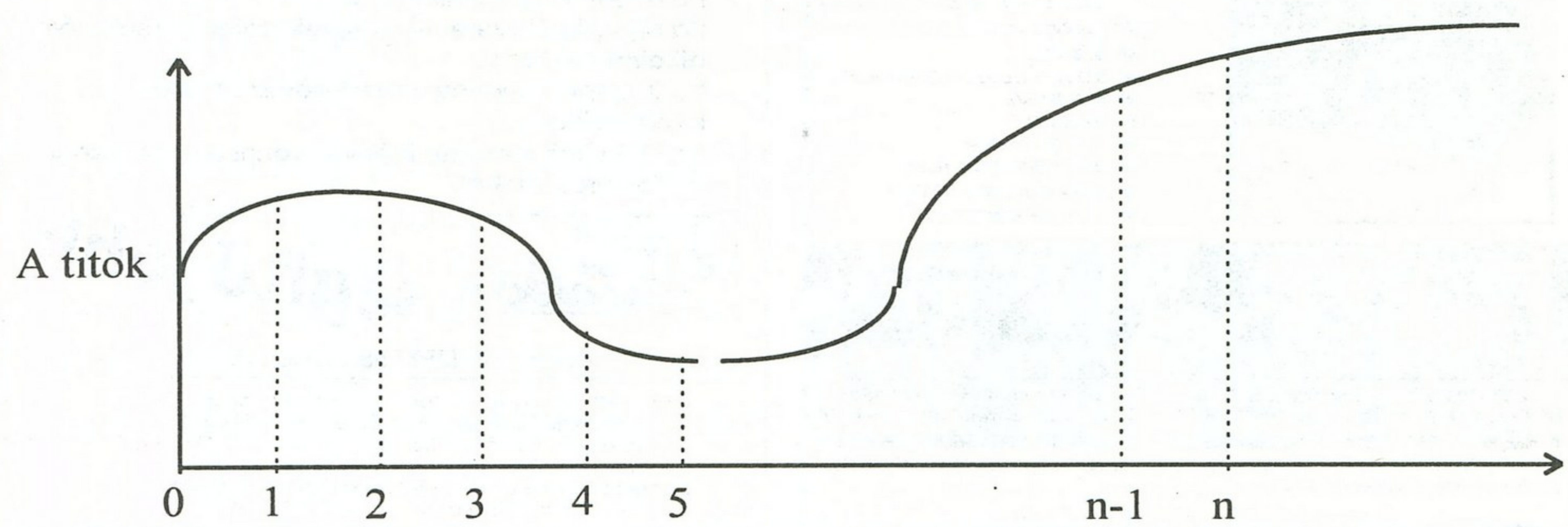
A titokmegosztás matematikai módszerei mindkét problémát megoldják. A legismertebb módszert Shamir találta fel, s a Lagrange-féle polinominterpoláción alapul. Ennek lényege az, hogy egy $k-1$-edfokú polinom együtthatóit vissza tudjuk állítani, ha ismerjük a k különböző helyen felvett értékét. Legyen a titok a 0. helyen felvett érték (vagyis a konstans együttható), és legyen a többi együttható véletlenül választott. Ha n részre akarjuk bontani a titkot, akkor legyenek a titokdarabok a polinom $1, 2, \ldots n$ helyen felvett értékei. Ha az n érték közül ismerünk k darabot, s azt, hogy az adott értéket hol vette fel a polinom, akkor a polinom együtthatói visszaállíthatóak. Így a 0. együttható is, ami a titok.

Az olyan sémákat, ahol n titokbirtokos közül bármelyik k vissza tudja állítani a titkot, küszöbsémáknak nevezzük. Nézzük meg egy példán, mit tud nyújtani ez a módszer.

A küszöbséma módszer

Tegyük fel, hogy a rendszergazda belépési jelszavát szeretnénk felosztani.

Meghatározhatjuk azt, hogy hány személy között akarjuk a titkot szétosztani, legyen ez mondjuk 5. Meghatározhatjuk azt is, hogy az 5 ember közül hány szükséges a jelszó



A Shamir-féle küszöbséma matematikai háttere

Válasszunk egy *p* prímszámot, amely nagyobb, mint a titok (ha a titkot bitekkel felírt számnak tekintjük). Legyen *(n)* a kiosztandó titokdarabok (árnyékok) száma, és legyen *m* árnyék szükséges az M titok rekonstruálásához. Véges test feletti algebrai egyenleteket használunk, ami azt jelenti, hogy a számokat modulo p tekintjük, s az alapműveleteket modulo p végezzük el. A titok megosztásához generáljunk egy tetszőleges *m–1*-edfokú polinomot a fenti p elemű test felett. Tehát ha egy *(m,n)* megosztási sémát akarunk defniálni, akkor generáljunk egy m–1-edfokú

$$(a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \dots + a_1x + M) \bmod p$$

polinomot, ahol a p véletlen prím, nagyobb valamennyi generált együtthatónál. Az a_j (j = 1, m–1) együtthatók véletlenszerűen választottak, titkosak, és az árnyékok szétosztása után eldobhatók. M azonos a titokkal. A p prím nyilvánossá is tehető.

Az árnyékokat a polinom n különböző pontban kiszámolt helyettesítési értékeként nyerhetjük:

$$k_i = F(x_i) \bmod p \ . \quad (i=1,\dots,n)$$

Más szóval, az első árnyék lehet a polinom értéke az x = 1 helyen, a második árnyék lehet a polinom értéke az x = 2 helyen és így tovább. Ezzel lezártuk a feladat első részét, a titok szétosztását.

Mivel egy m–1-edfokú polinomnak m ismeretlen együtthatója van (most: a_{m-1}, ..., a_1 és M), bármely m árnyék segítségével felállítható m lineáris egyenlet, amelyben az együtthatók az ismeretlenek. Ugyanekkor m–1, illetve ennél kevesebb árnyék nem elegendő a lineáris egyenletrendszer megoldásához, míg m-nél több árnyék már redundanciához vezet. Ha tehát megoldjuk a lineáris egyenletrendszert (például Gauss-eliminációval), akkor a megoldások között szerepel a nulladfokú tag M együtthatója is. Így megoldottuk a feladat második részét is, visszaállítottuk a titkot. Vegyük észre, hogy nincs szükségünk az összes együtthatóra, csak az M-re. Ez az ötlet gyorsíthatja az egyenletmegoldást.

Egy rövid példán, amelyben n és m értéke nem nagy, szemléletesen bemutatható a fenti séma és megoldása.

Példa egy (3,5) titokmegosztási sémára, amelyben bármely három résztvevő együttesen rekonstruálni tudja az M titkot.

Legyen a titok M = 11. Legyenek a másodfokú polinom véletlenszerűen választott együtthatói 7 és 8, a véges test elemszáma pedig 13. Tehát a következő másodfokú polinomot generáltuk:

$$F(x_i) = (7x^2 + 8x + 11) \bmod 13$$

visszaállításához. Legyen ez, mondjuk, 3, vagyis ez egy (3,5) küszöbséma. Ekkor két titokdarab elvesztése vagy megrongálódása esetén a jelszó még visszaállítható. Ugyanakkor információelméletileg bizonyítható, hogy bármely két titokdarab együttesen is nulla információt hordoz a titokról, vagyis két titokdarab ismerete esetén is éppen annyi a szóba jöhető jelszavak száma, mintha nem tudnánk semmit. Ezt úgy is meg lehet fogalmazni, hogy a titokdarabok birtokosai (a titokdarabokat árnyéknak is nevezik), ha háromnál kevesebben vannak, akkor ugyanannyit tudnak a titokról, mint a kívülállók. Az ilyen sémákat méltán nevezik tökéletes sémáknak.

Igaz az is, hogy ha a séma tökéletes, akkor a titokdarabok mérete legalább akkora, mint a titok (méreten a leíráshoz elegendő bitek számát értjük). Ugyanis két titokdarab-birtokos összesen is nulla információval rendelkezik a titokról, vagyis a bizonytalanság megegyezik a titok méretével. Nyilvánvaló tehát, hogy a harmadik titokdarabnak legalább annyi bit információt kell hordoznia, mint a titok mérete.

Egy másik megközelítés

Egy Blakley nevű úr másképp közelítette meg a problémát. Konstrukciója szellemes és szemléletes. Legyen a titok egy pont (ennek koordinátáival) a háromdimenziós térben. Legyenek a titokdarabok a pont vetületei az (x,y), (x,z), (y,z) tengely által meghatározott síkra. (Ez az eredete az árnyék elnevezésnek.) Egy ilyen pont egy egyenest határoz meg, amelyen a titoknak rajta kell lennie. Tehát bármely két titokbirtokos vissza tudja állítani a titkot, ez egy (2,3) küszöbséma. Ez a séma nem tökéletes, hiszen a titokdarab birtokosa tudja, hogy a titok melyik egyenesen van, míg egy kívülálló nem. Természetesen egyéb paraméterű sémák is készíthetők a tér és az alterek dimenzióinak megfelelő megválasztásával.

Megjegyezzük még, hogy itt általában nem euklideszi, hanem véges projektív térben dolgozunk.

Már a Shamir-séma is figyelembe tudja venni azt, ha a titokdarab-hordozók nem egyformán fontosak. Osszuk az árnyékok hordozóit két csoportra, az első csoport tagjai kapjanak egy titokdarabot, míg a második csoporthoz tartozók kettőt. Ha négy titokdarab szükséges a visszaállításhoz, akkor a második csoportból bármelyik kettő, az első csoportból bármelyik négy személy állíthatja vissza a titkot, illetve a harmadik lehetőség az, hogy az első csoportból kettő, a második csoportból egy titokdarab-birtokosra van szükség a visszaállításhoz.

Általában igaz: ha egyesével meghatározzuk azokat a részhalmazokat, amelyek jogosultak a titok visszaállítására, akkor konstruálható olyan titokmegosztási séma, amely ezt teljesíti. Ekkor például az A egyedül, a B, C, D hármas együtt, és a D és E kettős is képes a titkot visszaállítani. Egyetlen természetes feltételt szükséges teljesíteni: ha egy halmaz jogosult a titok visszaállítására, akkor minden olyan halmaz is, amely azt tartalmazza. Ha három ember jogosult a titok visszaállítására, akkor ha egy negyedik csatlakozik hozzájuk, a csoport továbbra is képes a titok visszaállításra.

Felhasználási területek

A titokmegosztást elsőként kulcsellátó rendszerekben alkalmazták. Sok felhasználót az tart vissza a rejtjelzés alkalmazásától, hogy fél a rejtjelzéshez használt kulcs elvesztésétől. Titokmegosztás alkalmazásával a rejtjelző kulcs visszaállításához szükséges mesterkulcs több részre osztható, különböző helyeken, különböző személyek felügyelete alatt őrizhető.

Megoldható egy digitális aláírás jogosultságának „szétosztása" is. Jó példa erre az, ha egy bankban egy nagy értékű átutalást bármely két alelnök együttesen írhatja alá. Ez jobb, mintha külön-külön írnának alá, hiszen ha a bankon kívül

valaki ellenőrizni szeretné a tranzakciót, akkor őt nem érdekli, hogy éppen ki volt az aláíró, sőt az aláírók változása kifejezetten zavarja, ő a bank egyetlen aláírását szeretné látni.

A titokmegosztás segíthet olyan esetekben, amikor olyan emberek, csoportok dolgoznak együtt, akik nem teljesen bíznak egymásban. Vegyünk példának egy mérőműszert, amely valamilyen orosz–amerikai hadászati megállapodást ellenőriz, és az eredményeket egy kulccsal rejtjelezi. Azt szeretnék elérni, hogy csak együttesen juthassanak az eredményekhez. Ez egy egyszerű (2,2) sémával megoldható. (Ebben az esetben megoldható egyszerűbben is.) Életszerűbb feltevés, hogy mindkét oldalon egy-egy bizottság dolgozik, s mindkét oldalról legalább két-két ember kell a kulcs visszaállításához. Ha még azt is elvárjuk, hogy a bizottságokból bármelyik két ember alkalmas legyen erre a feladatra, akkor segíthet egy olyan titokmegosztási séma, ahol a küldöttségek árnyékaiból bármelyik kettő egy egyenest határoz meg a síkon. A két küldöttség egyeneseinek metszése adja a titkot szimbolizáló síkbeli pontot.

„Árnyékolás"

A fenti példát leegyszerűsítve azt is mondhatjuk, hogy nagy értékű, rejtjelezett formában tárolt adatok kulcsának titokmegosztással történő védelme a titokmegosztási módszerek gyakori alkalmazási területe. Ma, amikor a hálózati kommunikáció minden formája rohamosan terjed, és gomba módra szaporodnak az Internet szabványait alkalmazó privát hálózatok, az informatikai rendszer biztonságának megteremtéséhez a titokmegosztás biztonságos kiegészítő eszközöket nyújthat.

A „szétosztott" öt árnyék a következő:

$k_1 = F(1) = 7 + 8 + 11 \# 0 \pmod{13}$

$k_2 = F(2) = 28 + 16 + 11 \# 3 \pmod{13}$

$k_3 = F(3) = 63 + 24 + 11 \# 7 \pmod{13}$

$k_4 = F(4) = 112 + 32 + 11 \# 12 \pmod{13}$

$k_5 = F(5) = 175 + 40 + 11 \# 5 \pmod{13}$

Rekonstruáljuk az M titkot tetszőleges három árnyékából. Legyenek ezek például k_2, k_3 és k_5. Ekkor a következő lineáris egyenletrendszert kapjuk:

$k_2 = 3$ esetében: $a * 2^2 + b * 2 + M = 3 \pmod{13}$,

$k_3 = 7$ esetében: $a * 3^2 + b * 3 + M = 7 \pmod{13}$,

$k_5 = 5$ esetében: $a * 5^2 + b * 5 + M = 5 \pmod{13}$,

azaz

$4 * a + 2 * b + M = 3 \pmod{13}$

$9 * a + 3 * b + M = 7 \pmod{13}$

$25 * a + 5 * b + M = 5 \pmod{13}$

A három egyenletből álló egyenletrendszerünk megoldása: a = 7, b = 8 és M = 11. Valóban visszanyertük az M titkot.

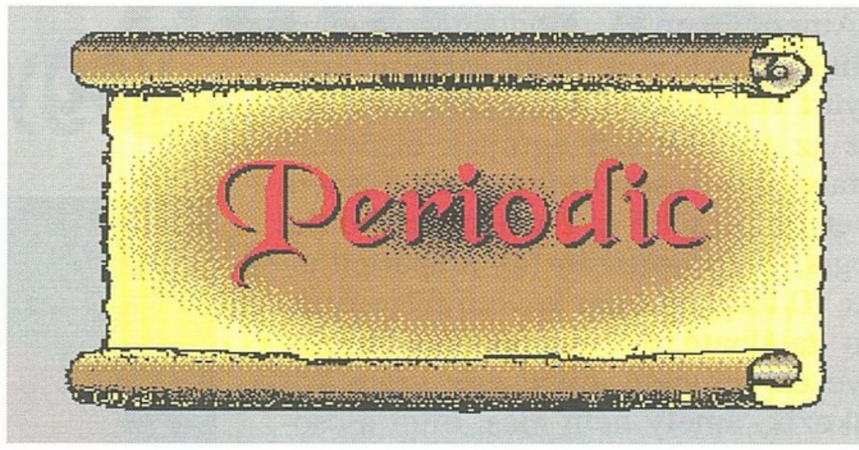
A sémát meg is csináltuk, maximum 1024 bites titkot tud szétosztani. Egy implementáció természetesen bonyolultabb az elméleti modellnél, ellenőrizni kell a titkok integritását, adminisztrálni kell a titokdarabokat. Ez a programrendszer alapvetően a fentiek szerint működik.

Papp Pál — Vecsera Ildikó

(A szerzőktől a programra vonatkozó információ a papp@mail.datanet.hu címen kérhető.)

Ha másképp nem megy...

„Elemi” ismeretek



Mint mondják: „Ismétlés a tudás anyja”. A Periodic program segíti az elemek és vegyjeleik bevésését — jelölve azok rendszámát is a panel felett.

A program folyamatosan, a mellékelt két ASCII fájl (.CHM) alapján ábécérendben sorolja fel az elemeket és vegyjeleiket. A felsorolás rendezve kérhető: az elemek neve szerint (ez az alapbeállítás), de vegyjeleik alapján is. Ezt a program futása során is meg lehet változtatni a gombok, illetve a Ctrl + E vagy a Ctrl + V billentyűk segítségével.

Szintén kapcsolható a hangjelzés az ikonon, illetve a Ctrl + H billentyűvel. A programot vagy az ESC billentyűvel, vagy az Alt + F4 billentyűkombinációval lehet megszakítani.

Az F1 billentyűvel kérhető egy rövid segítség a program futása során, de egérrel a „?” ikonon kattintva is. Egy további ikon, illetve a Ctrl + K billentyűkombináció szerepe az elemcsoportoknak megfelelő kiíratás. Az opció bekapcsolása a /K kapcsolóval végzett indítással is megoldható. Bekapcsolás után az elemcsoportokat jelző színek a képernyő alsó részén is megjelennek, és a csoportok rövid „tartalomjegyzéke” is kiírásra kerül. Ez utóbbi kiírás a program futása során az Alt + K billentyűkombinációval kapcsolható ki/be, de csak akkor, ha a csoport szerinti háttérjelölés aktív. Hasonlóan kapcsolható egérrel, a 10. csoport színgombjára kattintva. A kapcsolás megoldható a sorolást végző nyilakkal majdnem egy szintben mozgatva az egeret.

Hogyan kell kezelni?

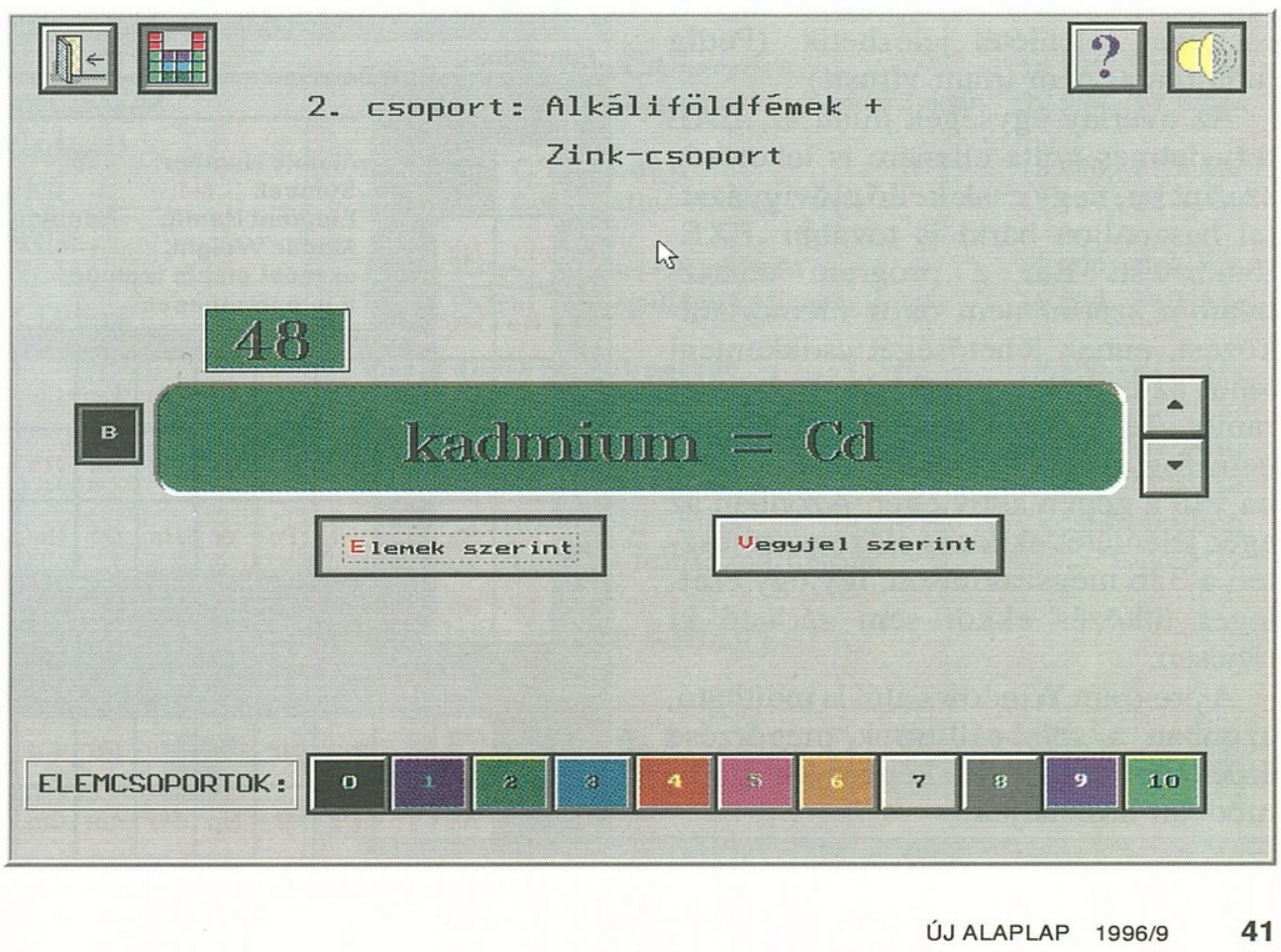
A nyilak közül a jobbra és a lefelé mutató az ábécé szerint előre, a balra és a felfelé mutató pedig visszafelé pörgeti a névsort. A program induláskor a bekapcsolt Numlock állapotot feloldja. Ezzel a numerikus billentyűk is használhatók kurzorként. Kilépéskor lehetőség szerint visszaállítja az eredeti állapotot. A program egy további lehetőséget is biztosít a csoportok áttekintésére. Ekkor az egérrel az „ELEMCSOPORTOK” feliratra kattintva az bekapcsolódik, miközben a középső feliratpanel is színt vált.

Ezt követően az alsó gombsor tagjainak bekapcsolása is lehetővé válik. Azokon az egérrel kattintva a felső mezőben megjelenik a megfelelő csoport rövid tartalomjegyzéke. A nyilak funkciója ez esetben is azonos az előbb említettel. Az üzemmódot az ESC billentyűvel, illetve az „ELEMCSOPORTOK” feliratra kattintva, vagy a jobb egérgombbal lehet elhagyni. Az itt megjelenő súgó csak ezt az üzemmódot támogatja. Ezt az üzemmódot egér nélkül az Alt + E billentyűkombinációval is bekapcsolhatjuk.

Adott elemet megkereshetünk a vegyjele szerint az Alt + B billentyűkkel, illetve egérrel a ’B’ gombon kattintva. De kereshetünk a rendszáma alapján is az Alt + R billentyűkkel, illetve a bal egérgombbal a rendszám ablakában kattintva. Ugyanitt a jobb egérgombbal kattintva (vagy Alt + S) az elemeket rendszám szerinti sorrendbe rendezhetjük. A visszatéréshez az ábécérend szerinti soroláshoz ugyanez a kapcsolás alkalmazható.

Miheztartás végett...

A programban alkalmazott csoportosítás a Römpp-féle kémiai lexikonban találtaknak megfelelő, nem jelölve külön színekkel az alcsoportokat. Ez alól kivételt képeznek 3. alcsoport „alcsoportjai”, a lantanidák és az aktinidák. Ezeket a 9., illetve 10. csoport tartalmazza. Az elemek neveinek, vegyjeleinek írásmódja szintén a fenti mű alapján került be a szövegfájlokba. Kivételt képeznek a 104-es rendszámú és annál nehezebb elemek. A 104-es elem kurcsatóviumként szerepel a program adatait tartalmazó fájlban, bár nevére P. W. Atkins „A periódusos birodalom” című, közelmúltban megjelent könyve dubniumot ad meg. A 105—109-es elemek nevét, rendszámát szintén e mű alapján tartalmazza a program. De ha az itt megadott név, illetve vegyjel nem megfelelő, ezt az ASCII fájlok átírásával módosítani lehet, viszont a csoportok rövid „tartalomjegyzéke” továbbra is a régit fogja tartalmazni. (Ha valamiben tévedek, szívesen kijavítom.)



Amennyiben a módosítás mellett döntünk, akkor a többi paraméterrel együtt végezzük a módosításokat. Ehhez az említett szövegfájlok első sora tartalmazza a csoportok számát. Ez nem lehet 10-nél több és 2-nél kevesebb (ha mégis, a csoportszám az alapértelmezett 10 lesz). Az ezt követő felsorolásban található az elemek rendszáma, neve és vegyjele, majd egy szám következik, amely nem más, mint a csoportot jelölő szám. Praktikus okokból a háttérpanelben kért szín kódja. Ha ez a szám a csoportok számánál nagyobb, a program az utolsó csoportba tartozónak fogja tekinteni. Az elemek neveinek írásmódja szintén módosítható, de a hosszú ékezetes kis-, illetve az 'É' kivételével a nagybetűknél a ku"riumnál talált ékezetjelölést szükséges használni (A', I', O', o", O", U', u", U").

Végül a programról

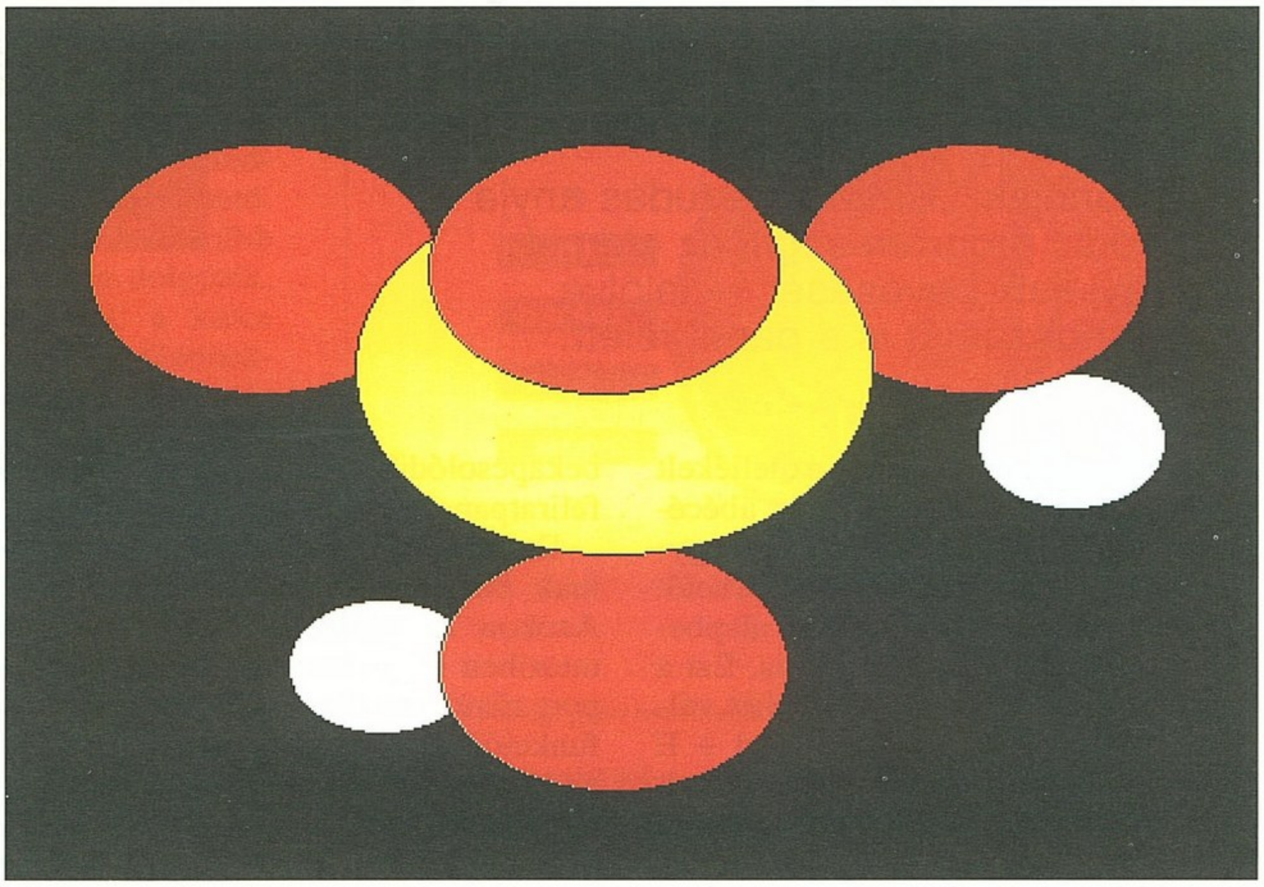
Borland Turbo Pascal 7.0 környezetben készült, helyenként objektumba (.OBJ) fordított Assembler, illetve Turbo C betétekkel. A PERIOD.EXE programfájl DOS alatt fut real üzemmódban, memóriaigénye azonban elég jelentős. Probléma nélküli futásához kb. 500 Kbájt szabad DOS memória kell. Támogatja a DOS-EMM használatát, ha van ilyen. Enélkül csak a hagyományos memóriában dolgozik. A program beépített overlay-egységeket is tartalmaz a futás közben fellépő alap-memóriaigény csökkentésére. Ezek az overlay-egységek lehetőség szerint szintén az EMS-t használják. Ebből is kb. 200 Kbájtot. Valószínűleg ez utóbbi okozza, hogy a Norton Antivirus Autoprotect régebbi verziói az első futtatáskor vírus esetleges jelenlétét jelezhetik. (Pedig tudtommal nem írtam vírust!)

Az overlay-egységek miatt az .EXE terjedelmes volta ellenére is lehetőség szerint ne, vagy csak kellő elővigyázattal használjon bárki is további .EXE-tömörítést. Bár a program tapasztalatom szerint nem okoz memóriaütközést, ennek lehetőségét csökkenteni lehet az egér kikapcsolásával. A programot /M kapcsolóval indítva szintén nem fog egeret használni. Akkor sem, ha van a gépen aktív egér. Azonban az egér jelenlétét ekkor is előzetesen teszteli a 33h megszakításon, így egy esetleges ütközés ekkor sem zárható ki teljesen.

A program Windows alól is indítható, azonban a színbeállítások megőrzése érdekében csak teljes képernyős üzemmódban használjuk.

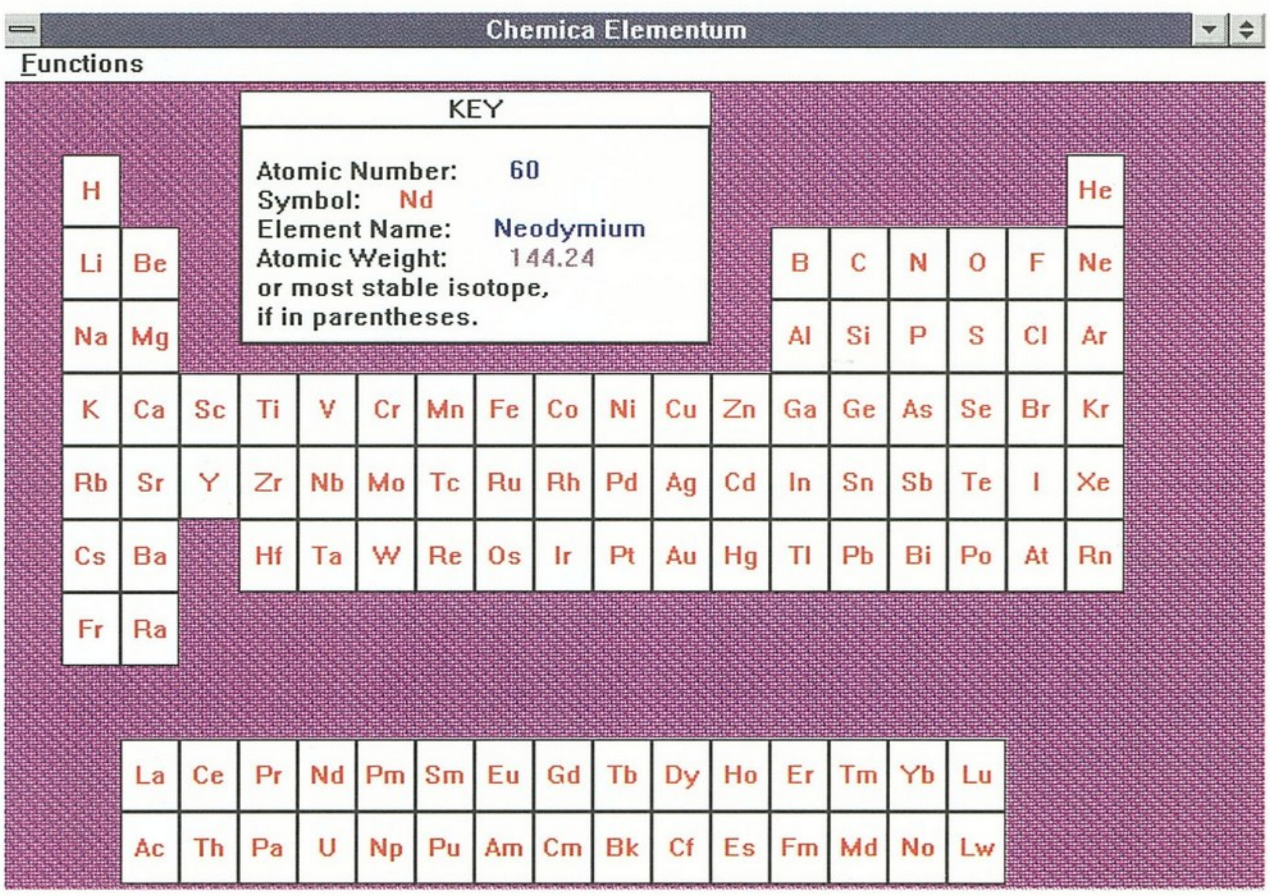
Simay Endre István

„Vegyes" anyagok a lemezen



A lapot (és a programokat) az élet szerkeszti… A szomszédos hasábokon olvasható cikkhez tartozó Periodic programot szerzője eredetileg Chemic névre keresztelte, csak amikor egy ugyanilyen nevű shareware-be ütközött, akkor változtatta meg Periodicra. A Chemic for Windows demója egyébként a periodikus táblázatot mutatja be, az egérkurzor alatti elemekről adva bővebb információt is.

S hogy a kép még teljesebb legyen, egy Chemical nevű shareware szoftver is előkerült (bár az meg Chem lehetne, mert így hívják az .EXE fájlt). A Chemical játéknak sem rossz: atomokból tudjuk vele összerakni a vegyületek molekuláit, és a modellt (lásd alábbi képünket) forgathatjuk, sokodalúan elemezhetjük stb. Mivel e programok jól kiegészítik egymást, mindhármat ráraktuk a lemezmellékletre. Az oktató célzatú kémiai „játékok" egyrészt az új tanév megkezdésére emlékeztetnek, másrészt előrevetítik jövő havi témánkat: az „edutainment" munkacíme TANJÁTÉK, ami egyúttal magyarítási kísérlet is. Tessék ízlelgetni!



INFORMÁCIÓKÉRÉS: 0904 ▶



AZ INTERAKTÍV MÉDIÁK MAGAZINJA

Multimédia Magazin Mindenkinek

Az idei második szám megjelent június végén

A tartalomból:
Mérlegen a hazai lexikon-CD-k, Monitorok óriástesztje, Slide show a karosszékből Portfolio Photo-CD-vel, Trükkök a CorelDRAW-val, Video a multimédiában, Audio-CD-válogatás, Ismerkedés a CompuServe-vel, Hálózati multimédia-fejlesztők és bemutatkozik többtucatnyi CD-újdonság.

A CD-MELLÉKLETEN

Mallorcai utazás, Soproni körkép, Lánchídi hangulatok, Dolák-Saly Róbert videoklipje, Az év fotói 1995, Multimédia-iskola, A tokaji bor apoteózisa, Új hazai CD-k demói és shareware csokor.

A CD Panoráma megjelenik évente négyszer, előfizethető a Computer Panoráma Kiadónál.
cím: Computer Panoráma Kiadói Kft.
1091 Budapest, Üllői út 25. II. emelet
Telefon: 218-3011/302, fax: 217-2646
Teljes éves előfizetés esetén ajándék CD-box!

MEGRENDELŐLAP
Előfizetéssel megrendelem 1996-ra a CD Panorámát
(négy szám ára: 3920 Ft, az idén még hátralévő két szám ára: 1960 Ft)

Név: ....

Postacím: ....

Telefon: ....

Olvasható aláírás: ....

LAPJAINKKAL CÉLBA TALÁL

INFORMÁCIÓKÉRÉS: 0904 ▶

Powered By DATAFLEX

Újdonságok a DataFlex világából!

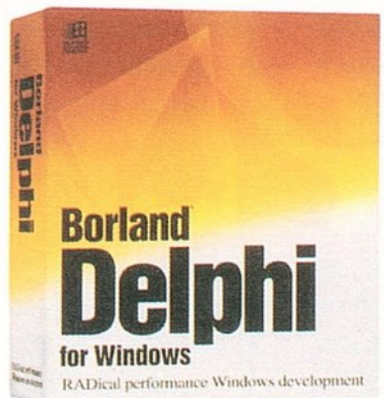
- DataFlex 3.1 DOS, OS/2, Unix környezetben
  Új szolgáltatások:
  Tranzakciókezelés Novell TTS-sel
  Beépített Application Framework
  Új utasítások
  Gyorsabb objektum- és indexkezelés
  DFIMPORT utility
  Adatbázis-adminisztrátori funkciók
- FlexQL 2.0, WINQL
- Új, **2.3b runtime**, gyorsabb indexkezelés, TTS-támogatás, 2000-re egyszerű megoldás
- DataFlex hírek, újdonságok az **INTERNET**-en a **DataFlex Klub** támogatásával

OOP kézikönyv és leírás magyarul is kapható!
Package-ek, utilityk magyar nyelven is!
Tanuláshoz segédanyagok rendelhetők.

NEXT Software Kft

Budapest XI., Andor u. 60. Tel.: 208-4643

INFORMÁCIÓKÉRÉS: 0917 ▶



Borland

legalizációs kampány

1996. szeptember 30-ig

az alábbi termékekre

upgrade alap bemutatása nélkül:

Delphi 1.0 Desktop C/Upgrade	**14.900,- Ft + Áfa**
Delphi 1.0 Desktop Legalization licence 2 user	**19.900,- Ft + Áfa**
BC++ 4.52 C/Upgrade	**16.000,- Ft + Áfa**
BC++ 4.52 Legalization licence 2 user	**19.900,- Ft + Áfa**
Turbo Pascal 7.0 Dos C/Upgrade	**14.000,- Ft + Áfa**
Turbo Pascal 7.0 Legalization licence 2 user	**19.900,- Ft + Áfa**
Visual dBase 5.5 C/Upgrade	**9.900,- Ft + Áfa**

Mindegyik csomag teljes dokumentációt, adathordozókat, licencet és regisztrációs kártyát tartalmaz. Mind a szoftvercsomagok, mind a licencek a későbbiekben upgrade alapként felhasználhatóak!

Ezt kár lenne kihagyni ...

SilVert

Borland Delphi-Szoft **Delphi-Szoft**
1085 Budapest, Horánszky utca 26.
Telefon: 138-4144, fax: 118-0915
Internet: http://www.delphi.hu

INFORMÁCIÓKÉRÉS: 0907 ▶

MoBiDic

A gépi szótárak új koncepciója

A szótárprogramok — még a legismertebbek is — megpróbálnak a lehetőségekhez képest a legjobban hasonlítani papír társaikra. Vajon tényleg erre van szükség a számítógépes környezetben történő fordításnál? És ha nem, milyen módon lehetne jobban támogatni a fordítási folyamatot? Ha pedig készül egy ilyen új típusú rendszer, használható-e a fordításon kívül másra is?

A szótár nem más, mint összefoglalása annak a tudásnak, amely a szavak különböző beszédhelyzetekben adott használatából alakult ki. Mint minden összefoglaló mű esetében, természetesen itt is előfordul olyan eset, hogy a szótárban talált információ felülbírálandó, vagy legalábbis kiegészítendő. Más szóval, a szótár sohasem tartalmazhat minden ismeretet, csak lehetőleg sokat. Ezért a fordítónak sohasem lehet elegendő szótár a környezetében, mindig kell a fordítói intuíció, ami nem elsősorban a szavak, kifejezések, hanem sokkal inkább a fordítási szokások ismeretén alapul. Ennek a tudásnak a támogatása pedig éppen nem a kivonatolt, összefoglalt szótárszerű információval, hanem rengeteg példával (akár a különféle szótárakban összefoglaltakkal, akár a meglevő fordításokban találtakkal) a legcélszerűbb. Ehhez pedig a hagyományos szótártípusoktól eltérő struktúrájú szótárakra van szükség.

Mitől intelligens egy szótár?

„A szótár — akár könyv alakban kiadott, akár számítógépes — önmagában annyi intelligenciát tartalmaz, amennyit a szerzők, szerkesztők belevittek" — mondhatnánk, ha nem éreznénk, hogy sokszor használat közben derül csak ki egy-egy mű valódi értéke. Például jól vagy éppen nem jól tudunk keresni benne, vannak vagy nincsenek jó mutatói, indexei, könnyen vagy nehezen találhatók meg benne a kifejezések, és még ki tudja, hány szempont adja meg a szótár használhatóságát.

Számítógépes megvalósításkor könnyen érvényesíthetők olyan szempontok, mint a fordítás közben való használhatóság, a működési sebesség növelése vagy a kézikönyvek nézegetési, tallózgató stílusának a támogatása — ettől azonban még nem mondhatjuk, hogy növekedne a szótárak intelligenciája.

A könyv alakban történő megjelenítés elvileg képtelen néhány olyan dologra, amit pedig a gépi megvalósítás lehetővé tenne. Ilyenek például a keresés nyelvi pontossága, a bővíthetőség és a párhuzamos működtethetőség. A könyvszótárakat utánzó számítógépes szótárak ugyancsak adósak ezzel. Márpedig úgy tűnik, hogy éppen a fenti szempontok érvényesítése jelenthetne olyasmit, amit a gépi szótárak „intelligenciájának" nevezhetnénk. A következőkben a gépi szótáraknak ezeket a lehetséges előnyeit vesszük sorra.

A karakterkészletek megjelenítésének szabadsága

Természetes követelmény, hogy minden olyan nyelvnek a karakterkészlete megjeleníthető legyen, amely legalább egy szótárban előfordul. Így Európában minimális követelmény, hogy a nyugat-európai, a kelet-európai, a cirill, a görög, a török és a balti kódlapokon szereplő betűket használó nyelvek szavai az adott nyelv pontos helyesírásával jelenjenek meg.

Nyelvfüggő ábécérend

Mivel az egy kódlapon szereplő karakterek nyelvenként más szabvány szerint kerülnek ábécésorrendbe, minden előforduló nyelvhez mellékelni kell egy olyan információt, amely ezt a sorrendet leírja.

Szavak, kifejezések intelligens keresése

A keresés kérdéskörén belül érdemes különválasztani a szavak keresésébe bevihető intelligenciát a többszavas kifejezések kezelésétől.

1. Szótári tövek előállítása az aktuális alakból

A szótári címszavak nyelvenként és szófajonként különbözőek: a magyarban a névszók alanyesetben, az igék egyes szám harmadik személyben állnak, de például a németben az igék szótári alapformája az infinitívusz. A szövegekben előforduló szóalakok (hiszen általában innen indul a szótárazási folyamat) először egy szótári alakokat előállító programmodulon kell, hogy átmenjenek, ezáltal minden bemenő szóra egy vagy több szótári tövet állítunk elő. Ha pl. a bemenő alak a KIMENTEK, akkor mind a KIMENT (excuse), mind a KIMEGY (go out) alak előáll keresendő tőként.

Effajta többletet csak az algoritmikus úton működő nyelvelemző rendszerek hozzáadásával lehet elérni. A szóalaktani (morfológiai) elemzés visszaadhatja a szó szófaját is, lehetővé téve a szócikk megjelenítésének leszűkítését csupán erre a szófajra.

2. Többszavas kifejezések intelligens keresése

A morfológiai elemzés, pontosabban a tőelőállítás (lemmatizálás) többszavas kifejezések esetén is működik. Ilyenkor azt várjuk el a rendszertől, hogy — ha kell, a keresett szavak sorrendjétől függetlenül — jelenítse meg mindazokat a szótárban szereplő kifejezéseket, melyekben a keresett szavak tövei előfordulnak. Tehát: a GET GOING keresőkérdésre mondjuk az „I've got to go" kifejezés is találat lesz, hiszen az eredeti kifejezésben szereplő mindkét tő, a GET és a GO egyaránt szerepel benne. Igazi előnye ennek a módszernek, hogy olyankor is megtalálja a program a kifejezést, amikor külön nem is kérjük tőle: ha például nem tudjuk, hogy egy kifejezés valamelyik elemén állunk, ráadásul a szó más személyben, számban, módban, időben stb. szerepel a szövegben, mint a kifejezésben.

Az is előfordulhat, hogy egy kevésbé jelentősnek tekintett tagjáról ismerünk fel olyan kifejezést, amely a „papír szótárakban" csak egy helyen — az ún. kulcsszónál — található meg. Míg nem ismertük fel magát a kifejezést, addig annak tagjairól sem lehet semmiféle elképzelésünk, minősítésünk. Például a „szem előtt tartom" kifejezés ilyen formában nincs meg a szótárban, rajta keresztül mégis megtalálja a rendszer az „ezt tartva szem előtt" és a „mindezt szem előtt tartva" kifejezéseket.

Egységes kezelői felület, korlátlan bővíthetőség

E kérdéskörön belül sorra veszünk néhány olyan lehetőséget, amelyet a MoBiDic szótárak biztosítanak a felhasználó számára.

1. Tetszőleges számú szótár egyidejű használata

A használatban lévő szótárprogramok egységes kezelői felületének kifejlesztése iránti igény valószínűleg sokakban felvetődött már, de a különféle szótári rendszerek készítői nehezen hozhatók egységes álláspontra a felület kialakítását illetően. Ugyanakkor a szótárprogramokat marketing okokból semelyik gyártónak nem érdeke közös nevezőre hozni, hiszen a „motor", a szótárak kezelését végző programrész így csak egyszer volna eladható, ezután a szótári adatbázisokat mint információtárakat, kiegészítő egységeket lehetne csak értékesíteni. Természetesen ez a felépítés előnyt is jelenthet, ha az egész rendszert már kezdetettől fogva modulárisnak tervezik.

2. Kész MoBiDic szótárak

A MoBiDic szótárak — technikai szempontból nézve a kérdést — adatbázisok, melyek működtetéséhez a MoBiDic alapprogram szükséges. A bennük szereplő nyelvi anyag részben már meglévő, könyv alakban is kiadott szótárak szavait, kifejezéseit tartalmazza, részben az éppen MoBiDic felhasználói (professzionális felhasználók, fordítók, fordítóirodák) által MoBiDic rendszerben készített szótárak, amelyek csak géppel olvasható formában léteznek, és természetesen ki vannak egészítve megfelelő morfológiai tudással.

3. „Rálátás" más létező szótárprogramok adatbázisaira

A MoBiDic szótári rendszer a fenti filozófiát alkalmazza, így belőle nem pusztán a hozzá kapható szótárak, hanem a más szoftvergyártók által forgalmazott szótárak is elérhetők — a programból való kilépés nélkül —, amennyiben ezt az illető partner is fontosnak tartja. (Jelenleg folyamatban van több, más gyártótól származó, piaci forgalomban lévő szótár ily módon történő bevonása a MoBiDic technológia alá.)

4. Felhasználói szótárak

A felhasználó maga is építhet szótárakat, melyeket a MoBiDic rendszer egyidejűleg kezel a megvásárolható — bináris formájú — szótárakkal. Ezzel a rendszer azt az érzést kelti, hogy a felhasználó a meglévő szótári anyagba írhat bele. Ezt nem tehetjük meg persze, sem technikai, sem szerzői jogi okokból. De nincs is rá szükség, hiszen a több szótár egyidejű használata lehetővé teszi az egyidejű megtalálást. Sőt, a rendszer a megtalált szó szótári eredetét is jelzi, így mindig pontosan tudhatjuk, hogy melyik szótárból való a találat.

Természetesen ezeket a szótárakat el is kell készíteni, majd — igény szerint — bővíteni, módosítani: új szavakat, kifejezéseket felvenni, másokat törölni stb. Ezeknek a műveleteknek a MoBiDic rendszer teljes támogatását adja, lehetővé téve külső eszközökkel készített szószedetek automatikus beemelését is, jelentősen felgyorsítva ezzel az adatbevitelt.

Hozzáférési lehetőség szövegkorpuszokhoz

A legjobb szótárak néhány példát is hoznak a szócikk végén, annak illusztrálására, hogy az adott szót, kifejezést hogyan szokás használni szövegkörnyezetben. Napjaink rendszereinek lehetőségük van egyre nagyobb szöveges adatbázisok, ún. szövegkorpuszok elérésére.

A MoBiDic szótári rendszer alaptétele, hogy a különféle szótártípusok egyidejűleg érhetők el. Ha a mondatpéldákat tartalmazó adatbázis önálló egység, annak megvásárlása esetén a szónak nemcsak jelentését, értelmezését, hanem mondatkörnyezeteit is megkaphatjuk (esetleg fordítással együtt), ahol a keresett szavak előfordulnak. Ezek megjelenítése mindössze annyiban tér el a szokásos fordítói vagy értelmező szótárakétól, hogy a forrásnyelvi oldalon nemcsak a címszó, hanem az előtte és mögötte lévő, megfelelő méretű szövegkörnyezet is megmutatandó. Ekkor a szövegkorpusz és a szótári információ egyazon eszközkészlettel kezelhető. Mivel egyre több szöveg (pl. újság, könyv, fordítás, levelezés stb.) érhető el számítógépek segítségével, érthető módon a nemzetközi hálózatokkal való kapcsolat lehetővé tétele a fejlesztés egyik fő iránya.

Multimédia-lehetőségek

A fordítástámogató rendszerek elsősorban az írott szöveg más nyelven írott szöveggé való átalakításához adnak segédeszközöket. Sokszor itt különül el az általános célú vagy épp a tanítást segítő szótári jellegű információ számítógépes támogatása a fordítástámogatástól. Ennek ellenére a fordítási munkának is van olyan területe, ahol a nem szöveges információ nagyon fontos lehet. Gondoljunk például egy autójavítási vagy repülőgép-összeszerelési kézikönyvre. Sokszor még nincs is megfelelője az adott terminológiának a fordítás célnyelvén, ugyanakkor a körülírás csak akkor pontos, ha tudjuk, mit kell körülírni. Ilyenkor elengedhetetlen a képi információ. Természetesen a nyelvtanítás sem nélkülözheti a képes vagy hangos szótárakat. Nos, a MoBiDic technológiával tetszőleges multimédia-objektum (kép, hang, film stb.) megjeleníthető a találati oldalon.

A fordítási folyamat teljes támogatása

Szövegek más nyelvre fordításához hosszú időn át azért nem volt megfelelő számítógépes eszköz, mert a gépi megvalósítás a könyvformátumot próbálta leképezni a számítógép képernyőjére. Márpedig a fordítást végző felhasználónak nem erre, hanem egy minél több „igazi" számítógépes szolgáltatásokkal ellátott, hatékony, valódi gépi eszköznek kellene lennie, melyben az elméleti nyelvészet eredményeitől kezdve a felhasználói viselkedés elemzéséig mindennek szerepet kell látszania. A fent vázolt dinamikus mintapélda-gyűjtemény legfontosabb erénye, hogy egy korábbi cikkemben említett sakkprogram-fejlődés mintájára (Új Alaplap, 1995/10) igen hatékonnyá teszi a géppel támogatott fordítást.

A géppel elérhető fordítások további nehézségeket vetnek fel: például a szöveg „szinkronizálásának", azaz a fordítandó és a lefordított nyelvi egységek (szavak, szerkezetek, mondatok, bekezdések) összepárosításának a problémáját. A fordítómemóriának nevezett szoftvereszközök (Trados, Transit, Translation Manager stb.) ezt általában mondatszinten végzik, de az igazán jó mintapélda-megtaláláshoz részszerkezetek és nagyobb egységek fordításait és eredetijeit is hasznos párba állítani.

A fordító egyébként más nyelvi eszközök használatát is igényelheti, pl. a nyelvi ellenőrzőkét, a szóellenőrzőkét és a mondatellenőrzőkét. Ez utóbbiból egy egész gépi fordítási irányzat nőtt ki: az irányított nyelvtan fogalma. Ez nem más, mint egy olyan nyelvtani ellenőrző, mely nem feltétlenül a hibás (vagy annak gondolt), hanem a számára nehezen fordítható mondatok újrafogalmazását kéri, rávezetve ezzel a felhasználót az olyan mondatok használatára, melyet a fordítórendszer már maga is képes helyesen lefordítani.

Megvalósítási kérdések — CD vagy nem CD?

A szöveges és szótári adatbázisok egyidejű használata megerősíti a természetes nyelvi szövegekre vonatkozó nemzetközi szabványjavaslatok figyelembevételét a rendszer továbbfejlesztésénél. (Ilyen az általános szövegleíró nyelv, a Standard Generalized Markup Language, rövidítve: SGML, valamint az erre épülő Text Encoding Initiative, a TEI.)

Egy másik, és a gyakorlatban is rögtön szembetűnő megvalósítási kérdés, hogy szabad-e a szótári adatokat kizárólag CD-n elérhetővé tenni. Gondoljuk csak el, hogy napjainkban a számítógépes információt egyre gyakrabban CD hordozza. Ha az ezeken levő — nem magyar nyelvű — anyag megértésével gondunk van, hívjuk, illetve csak hívnánk a szótárprogramot, ugyancsak CD-n van. Mivel CD-lejátszónk épp az információforrást tartalmazó CD-vel van tele, több CD egyidejű kezelésére alkalmas CD-tornyunk pedig feltehetően még nincs, a szótári információhoz a CD-k cseréjével, vagyis hosszadalmasabb és kényelmetlen úton juthatunk.

A fentiek miatt megkülönböztetjük a CD-t mint adathordozót, és a CD-t mint valós idejű eszközt. Az előbbi azt jelenti, hogy van egy korszerű információhordozónk, az utóbbi pedig azt, hogy a szótárhasználat a teljes CD-kapacitást elveszi a szótár működtetésével egyidejűleg futtatandó programoktól. Természetes következmény, hogy a csak olvasható CD-ken működő szótárak MoBiDic-éhez hasonló felépítését sem tennék lehetővé.

A (közel)jövő

A nyelvi szoftvereszközök és a modern információtechnológiák ötvözésével keletkezett rendszerek — ilyen a MoBiDic is — kétféle úton válhatnak még hatékonyabb eszközökké: az általános technológiai fejlődés és a nyelvi technológiák fejlődése útján.

A nyelvi szoftverek új generációja figyelni fogja a mondatkörnyezetet, így képes lesz az aktuális mondat és a megtalált kifejezések viszonyának kiértékelésére. Más szavakkal ez azt jelenti, hogy az adott szóra mint egy kifejezés részére csak akkor talál rá a rendszer, ha az adott mondatkörnyezetben a kifejezés többi része is megfelelően illeszkedik. Vagyis: ha pl. az „esedékesség napja" kifejezés áll egy számviteli szövegben, akkor a „nap" szón állva nem a DAY megfelelőt veszi elő a rendszer, hanem a környezetre jól illeszkedő „esedékesség napja", azaz DUE DATE kifejezést. Persze a „nap" szó esetében még az sem biztos, hogy a SUN vagy a DAY az előveendő megfelelő. Ha tehát a kifejezés a „nap heve", akkor SUN, ha pedig a „nap hőse", akkor DAY a fordításban megjelenő nap-ekvivalens.

Prószéky Gábor

Merre haladunk?

Grétsy a számítógépben

A mai rohanó világban az újságok, folyóiratok szerkesztőinek, tördelőinek percek alatt kell hasábokba, oldalakba rendezni a friss híreket. Emberi lektorálásra vagy nincs ott a megfelelő személy, vagy nincs elég idő. A gondok egy részét ugyan leveszi az ember válláról a gép. De azért vannak még hibák...

Ma már nemcsak a sokak által használt Word, WordPerfect szövegszerkesztőben használják a helyesírás-ellenőrző, tördelő és automatikus elválasztó programokat, hanem a „hivatásos" kiadványkészítőkben is, legyen az PageMaker, QuarkXPress vagy bármi más. Persze a jó szemű lektort aligha helyettesítheti valaha is a gép.

A magyar nyelvre kb. 5 éve jelent meg az első általánosan használható helyesírás-ellenőrző. Azóta a programok minősége sokat javult, pontosították a nyelvtani ismereteket, bővítették, javították a meglévő szótárakat. De a programok szolgáltatásai is bővültek: fejlődött a szöveg ellenőrzése közben végrehajtható javítások módja, és sok helyen automatikus elválasztással is kiegészítették az ellenőrzést.

Mekkora lehet a selejt?

A tapasztalatok alapján a magyarban a szóellenőrzés segítségével az elkövetett hibáknak mintegy 95%-a kiszűrhető. A maradék 5%-kal azonban semmiképpen nem lehet megbirkózni. Az angolul, franciául íróknak jobb dolguk van: náluk a szóellenőrző programok után az elkövetett hibáknak alig 2%-a marad kiszűretlen. A különbség abból adódik, hogy a magyar nyelvben sokkal több lehetséges szóalak van, mint az indoeurópai nyelvekben. (Az elválasztási hibákról most ne is beszéljünk, az külön téma.)

Tegyük fel, hogy helyesírás-ellenőrző programunk olyan tökéletes, mintha Grétsy László csücsülne gépünkben, egyik kezében a Helyesírási kéziszótár, a másikban a Magyar helyesírási tanácsadó, a harmadikban Elekfi László Magyar ragozási szótára, és ha még maradt szabad keze, és található mérvadó szótár, azt mind segítségül tudja hívni. Ha csak annyit tud, hogy minden szóról pontosan megmondja, helyes vagy nem helyes, akkor ne bízzuk rá magunkat a számítógépre.

Az a mondat például, hogy „A magasabb áru árú nem kel el" helytelen, bár minden szava helyes. Helyessé tehető, ha az „áru" és az „árú" szavakat felcseréljük — csakhogy a „helyesírás-ellenőrző"-nek nevezett program észre sem veszi a hibát, mert nem nézi meg a szó környezetét. (Nem is indokolt helyesírás-ellenőrzőnek nevezni, hiszen csak szóellenőrző.) A mondatkörnyezet figyelmen kívül hagyása persze más nyelvekben is hibákhoz vezethet,

például angolban sem mindegy, hogy mikor írjuk a „then" szót, mikor a „than"-t. Az angoloknak azonban elég bajuk van a szavak helyes leírásával is, így az angol nyelv nem is különbözteti meg élesen a két fogalmat.

Régebben a számítástechnikában is csak „spelling checker"-ekről, szószintű ellenőrzőkről beszéltek, később azonban megjelentek a szóhatárokon túllépő, mondatszintű elemző programok is. Ezeket a mondatszintű ellenőrzőket szokták „grammar checking" vagy „proof reader" néven emlegetni.

Szereptévesztés

A Word szóellenőrzőjébe sajnos kikapcsolhatatlan szolgáltatásként beépítették azt a funkciót, amely hibát jelez, ha ugyanaz a szó egymás után többször következik. Pedig angol nyelv is produkálhat formai szóismétlést: „She had had ...", és magyar szövegekben sem ritka a helyes szóismétlést tartalmazó mondat. Például: „Nem az az én kutyám, hanem a másik!" Egyes esetekben akár háromszor is megjelenhet ugyanaz a szó.

A szóismétlés kiszűrése sokkal inkább a mondatszintű ellenőrzés feladata lenne, nem a szóellenőrzésé. Megjegyzem, hogy a megadott példákban csak formai az azonosság. A magyarban az egyik „az" szó névmás, a másik névelő, az angol példa két igéjének pedig a funkciója más. Az ilyen finomabb különbségek azonban már csak a mondatelemzés során deríthetők ki.

A mondatok szerkezetének vizsgálata

A szintaxis határozza meg, hogy helyesnek tartunk-e egy mondatot. A gyakorlatban a szintaxist legtöbbször olyan „CF nyelvtannal" vagy annak módosított változatával próbálják leírni, amely környezetfüggetlen (Context Free, CF) generáló szabályok véges számú összességéből áll. Az ilyenfajta nyelvleírás lényege, hogy véges számú generáló szabállyal leírható a nyelv potenciálisan végtelen sok mondata.

Ezek a szabályok elemzésre is használhatók, hiszen megfordítva is értelmezhetjük őket: részekre bontható először a mondat, majd az egyes részek a többi résztől függetlenül tovább szeletelhetők. A programozási nyelvek fordításának gyakorlatából is széles körben ismeretesek azok az eszközök, amelyekkel az ilyen jellegű nyelvtanok könnyen kezelhetők.

Angolban az is jól megállapítható az ilyen egyszerű nyelvtanokkal, hogy melyik szó lehet az alany, melyik a ragozott ige. Ennek alapján a gép jónak találja a „The fly flies." vagy a „The flies fly." mondatokat, de fennakad a hibás „The flies flies." vagy a „The fly fly." mondatokon.

Ez a fajta leírás alapvetően a szavak, mondatrészek sorrendjére épít. Angol mondatok elemzésére azért is nagyon hatékony, mert angolban a mondat szerkezete szinte kizárólag a szavak sorrendjétől függ.

Persze a módszer használatához további információkra is szükség van: minden egyes szóról tudnunk kell például, hogy milyen szófajhoz tartozik, milyen raggal (esetleg más jellel) van ellátva, milyen ragozási típusba tartozik stb.

Nem elég tehát megállapítanunk, hogy a szó helyesen van leírva, hanem a szótárból vett információk alapján a szó alakjából további következtetéseket kell levonnunk. Angolban mindez kisebb probléma, de a magyarban komoly morfológiai elemzésnek kell megelőznie a szintaktikait.

Át lehet-e siklani az ismeretlen szavakon?

Angol szövegek elemzésében is problémát jelent, amikor ismeretlen szó bukkan fel a mondatban. Ha a szóelemző nem ismer fel egy szót, megkérdezi a felhasználót. Kérésre akár meg is jegyzi a szót, és a legközelebbi előfordulásnál már nem kíváncsiskodik.

A mondatok elemzésénél kicsit más a helyzet. Előfordulhat, hogy a mondat teljes elemzése ugyan sikertelen, de például a hibák többsége így is felderíthető. A „Koratte moziba mentem látok." mondat akkor is vizsgálható, ha a Koratte szóról csak feltesszük, hogy ismeretlen ugyan, de helyes. A felhasználót tehát fölöslegesen nem kell mindig háborgatni, legfeljebb közölni kell vele, milyen feltételezéssel élt a rendszer (például, hogy milyen szófajhoz tartozik az ismeretlen szó).

A szabályok száma és a szórend

A szabad szórendű nyelveknél gyakorlatilag is nehezen kezelhető problémát jelent a szabályok számának mérhetetlen megnövekedése. Gondoljuk el, mennyire felduzzasztja a nyelvtant, ha minden szabályt annyiféleképpen veszünk fel, ahányféle sorrendben értelmes kifejezés keletkezhet belőle.

A magyarban is vannak kemény szórendi kötöttségek. Nem keverhetjük össze például a szavakat abban a kifejezésben, hogy „sárga szárnyú kis liba". De már a „Péter szereti Katit" mondatban tetszőleges sorrendben írhatjuk a szavakat, és minden esetben helyes mondatot kapunk.

Nem véletlen, hogy az első szintaktikus elemzők, generátorok, fordítók tárgya nem a magyar volt, hanem a sokkal kevesebb szabállyal és sokkal egyszerűbben elemezhető angol, és a bonyolultabb szerkezetű indoeurópai nyelvek elemzői is jóval később készültek el.

A magyar mondattan formalizálása és gépi elemzése még keményebb dió, mint a magyar szavaké. Biztató kísérletek vannak ugyan, de bizonyára évek szívós munkája kell még ahhoz, hogy elkészüljön a magyar szintaxis átfogó, számítógéppel is kezelhető modellje.

Összefoglalva a következő kérdések vetődnek fel a magyar mondatok elemzésénél:

1. Szavak tulajdonságainak megállapítása.
2. Ismeretlen szavak kezelése.
3. A szabad szórend szabályainak kezelése.
4. A hiányos nyelvtani tudás alkalmazása.

Naszódi Mátyás

INFORMÁCIÓKÉRÉS: 0934 ▼

ÚJDONSÁG ! Csak nálunk!

CALDERA Network Desktop
Internet/Intranet/WWW Server

Linux alapu komplett megoldás, a vetélytársak árának töredékéért!

SOFT-WARE-STATION
SOFTWARE-EK ÉS SZAKKÖNYVEK PROFIKNAK

SERVER Funkciók:
Web (HTTP); Multi-domain Web; FTP; E-Mail; Gopher; USEnet News; WAIS; NFS; Dial-In Server (PPP/SLIP)
UNIX Server (NFS); Windows Server

CLIENT Funkciók:
Web (HTTP); FTP; E-Mail; Dial-Up; Management (SNMP)
Gopher; USEnet News (NNTP); WAIS; NFS
UNIX Client; NetWare 3.x/4.x Client; Windows Client

Routing Funkciók: Domain Name Server (DNS); NIS/NIS+; TCP/IP Routing; IPX Routing.

További termékek: WordPerfect+Motif; Internet Office Suite (WP+Motif+ZMail+NeXS Spreadsheet)

Telefon/fax: 371-0704. Fax Station: 181-3951

Teljes, 23.000 tételes szakkönyv adatbázisunk az Interneten!
http://sws.xco.hu ☆ BP., 1111 Karinthy Frigyes út 25.

A RT. radiant ®

Termelő, Szolgáltató és Kereskedelmi Rt.

1142 Budapest Kassai u. 84. Telefon:267-6770, 267-6771, 252-5125, FAX: 251-6850

Szerviz: 1142 Budapest Erzsébet királyné útja 53. Telefon: 252-1932, 251-1444

A világ egyik legnagyobb UPS gyártójának az

EMERSON
Computer Power

magyarországi dealere.
UPS-ek 250 VA-800 kVA-ig.

- **NEC optika, mikro;**
- **GN Elmi műszerek;**
- **műszerjavítások;**
- **Számítástechnika /hardver, szoftver/ forgalmazás, szervíz;**
- **monitor, UPS, tűs nyomtató, telefax, TV javítás;**
- **AUTÓRIASZTÓ forgalmazás;**
- **egyedi műholdvevők, CATV rendszerek.**

A KIMSOFT szeptemberi ajánlata

Windows'95 és alkalmazásai	
Windows'95 /Upgrade	29 996,-/15 900,-
ACCESS Win95 /Upgr.	**51 400,-/16 900,-**
Office for Win95 (magyar)	67 400,-/34 996,-
MS Office Prof. for Win95	**89 900,-/52 400,-**
Word for Win95 /Upgr.	51 400,-/16 900,-
CorelDRAW 6 (magyar)	**51 996,-/39 996,-**
Norton Commander Win95	10 996,-/ 5 900,-
Norton Utilities for Win95	**20 996,-/12 400,-**
Borland legalizálási akció	
Borland C++ v4.52	16 900,-
Borland Delphi v1.0	**15 900,-**
Turbo Pascal v7.0	14 900,-
Visual dBASE v5.5	**10 900,-**
Újdonságok, bevezető árak	
ABC Graphics Suite for Win95	**32 900,-**
AutoCAD LT Win95 /Upgr.	64 996,-/21 900,-
Kai's Power Goo	**14 900,-**
LightWave 3D v5.0	249 900,-/99 900,-
Propis 4.0 (Orosz helyesírás-ell.)	**13 400,-**
Symantec Café (Java fejlesztő)	21 900,-
CD-ROM-ok, játékprogramok	
7th Guest /Caesar II	2 900,-/6 400,-
Descent 2 /Duke Nukem 3D	**7 900,-/6 996,-**
F1 GrandPrix 2 /Mega Pack 5	7 900,-/7 400,-
Rebel Assault 2 /Settlers 2	**6 900,-/7 900,-**
Wing Commander IV /MYST	7 900,-/3 500,-
History of World /Le Louvre	**9 996,-/6 900,-**
Asterix az angoltanár 1. /2	5 400,-/5 400,-
Learn to Speak English /German	**21 400,-**
Nyelvmester (angol/német)	5 600,-
Angol-magyar hangosszótár	**7 900,-**
PIC-DIC (angol/német/francia) képesszót.	5 200,-

ABC Flowcharter v4.0 Comp. Upgr.	27 900,-
Adobe PhotoShop 3.0	**124 900,-/44 996,-**
Borland C++ 5.0 Dev. Suite	37 400,-
Corel WEB.Designer	**19 996,-**
Delphi Developer 2.0 Special	69 900,-
Clipper 5.3 (Akció!)	**Hívjon!**
Clipper kiegészítések	Hívjon!
Close Up 6.0 Dual Pack	**27 400,-**
CodeBase 6.0 (adatbáziskezelő)	82 400,-
CorelDRAW 5.0 /Up.	**56 400,-/36 400,-**
CorelDRAW 4.0 CD	19 996,-
10 db-os Corel Photo CD csomag	7 900,-
Delrina CommSuite 7.0	**20 400,-**
Explore 2.0 (Internetkezelő)	9 400,-
Fractal Design Painter 4.0	**73 900,-**
F-Prot 2.23 Prof. (antivirus pr.)	Hívjon!
IBM OS/2 v3 Warp (magyar is)	**Hívjon!**
Lotus Organizer 2.1 (magyar)	19 400,-
MS DOS 6.22	8 400,-
Netscape Navigator 2.0 Pers.	**10 400,-**
Novell NetWare 3.12 /4.1	Hívjon!
PageMaker 6.0 /Up.	**124 900,-/39 900,-**
Print Artist 3.0	13 900,-
QEMM 8.01 OEM (Akció!)	**9 996,-**
QuarkXPress 3.32 for Win.	124 900,-
Remove It 3.0	**9 400,-**
Uninstaller 3.5 (Windows takarító)	9 400,-
Visio 4.0 /Upgr.	**30 900,-/18 400,-**
Visual Basic 4.0 Prof.	Hívjon!
Visual C++ 4.0 Prof. Upgr.	**31 900,-**
Visual FoxPro 3.0 Prof.	74 900,-/46 900,-
WildCat 4.12 /5.0	**18 400,-/21 400,-**
WinFax Pro 4.0 / 7.0	13 400,-/16 900,-
WordPerfect 6.1 Suite CD Spec.	**32 900,-**

A közölt árak nem tartalmazzák a 25%-os áfát, és a helyszíni üzembehelyezés költségeit.

Részletes árjegyzékünket kérje telefaxon a faxbankból tone üzemmódban: 180-8611/1497#

KIM-SOFT Számítástechnikai és Kereskedemi Kft.
1112 Budapest, Hegyalja út 70. fszt. 2.
Telefon: 319-8973, 319-8967 Fax: 319-9760

INFORMÁCIÓKÉRÉS: 0929 ▼

BÉLYEGZŐKÉSZÍTŐ

P-touch StampCreator
bélyegzőszerkesztő programmal

Azonnal elkészül!
3 méret x 3 szín



WINDOWS alatt szerkeszthető

69.920 Ft+ÁFA

CÍMKENYOMTATÓ

P-touch PC
címkeszerkesztő programmal

A címke tartalmazhat:
szöveget, grafikát, vonalkódot



PC Windows vagy Macintosh környezethez

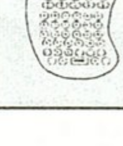
6-9-12-18-24 mm széles szalagok színes, vízálló, öntapadó címkék

63.400 Ft + ÁFA

DIT
DIGITÁLTECHNIKA
Győr, 9024 Mónus I. u. 19.
T/f: 96/414-411, 417-802
Budapest, 1149 Egressy út 5.
T: 30/463-657, T/f: 221-6779

brother
MÁRKASZAKÜZLET
SZERVIZ
DISZTRIBUTOR

FELIRATOZÓK



14.992 Ft-tól

SZÁMLÁZÓ PROGRAM



A számla fejléc, nyomtatványok, bizonylatok a felhasználó igénye szerint alakíthatók ki.
Széleskörű lekérdezési lehetőség.
Vevő-, szállító-, árunyilvántartás.
Windows 3.11/95 alatt használható.

Megfelel a 24/1995. (XI.22.) PM rendeletnek!

§ **9.984** Ft + ÁFA §

INFORMÁCIÓKÉRÉS: 0908 ▼

A mesterséges intelligencia „kísérleti” eszközei

A robotokkal „körbe járva”

A robotok — akárcsak a játékprogramok — remek lehetőséget adnak elméletek kipróbálására. Itt „pillanatok alatt” kiderül, hogy jó-e az elképzelés. Robotokkal nem csupán egyetemeken vagy gyárak laboratóriumaiban foglalkoznak, hanem élénk kutatás folyik például az amerikai haditengerészetnél is, a már ismertetett genetikai algoritmusok felhasználásával törekedve még jobb robotokat építeni, illetve még jobb programokat írni. Megpróbálnak a robotok eredményeiből géppel következtetni azokra a kritikus körülményekre, amelyekben mindenképpen ki kell próbálni a robotot. Lényeges, hogy minél kevesebb próbára legyen szükség, hiszen az idő pénz, és nincs lehetőség félmillió tesztre, mint például egy játékprogramnál.

Már az ókorban felmerült az igény, hogy az ember munkáját valamilyen géppel segítsük, bizonyos fázisokban az embert helyettesítsük. Ez azt is jelenti, hogy természeti környezetünkben kellene helytállnia egy gépnek, illetve a programjának.

A környezetben nem adhatunk meg mindent pontosan az utolsó milliméterig, grammig, így a programot nem csupán bizonyos állapotokra, hanem igen gazdag lehetőségekre kell felkészíteni. Elvégre nem állhat meg amiatt egy teljes gyár, hogy az egyik szerelőszalag fél centivel odébb állította meg a munkadarabot.

Sokan sokfajta feladatra

A robotikával foglalkozók számára igen fontos az input és az output kérdése is. Mi most nem foglalkozunk azzal, milyen érzékelőkre és műkezekre van szükség, hogy robotunk megfogjon és felemeljen egy tojást — anélkül, hogy elejtené vagy eltörné. (Pedig ez is érdekes téma.)

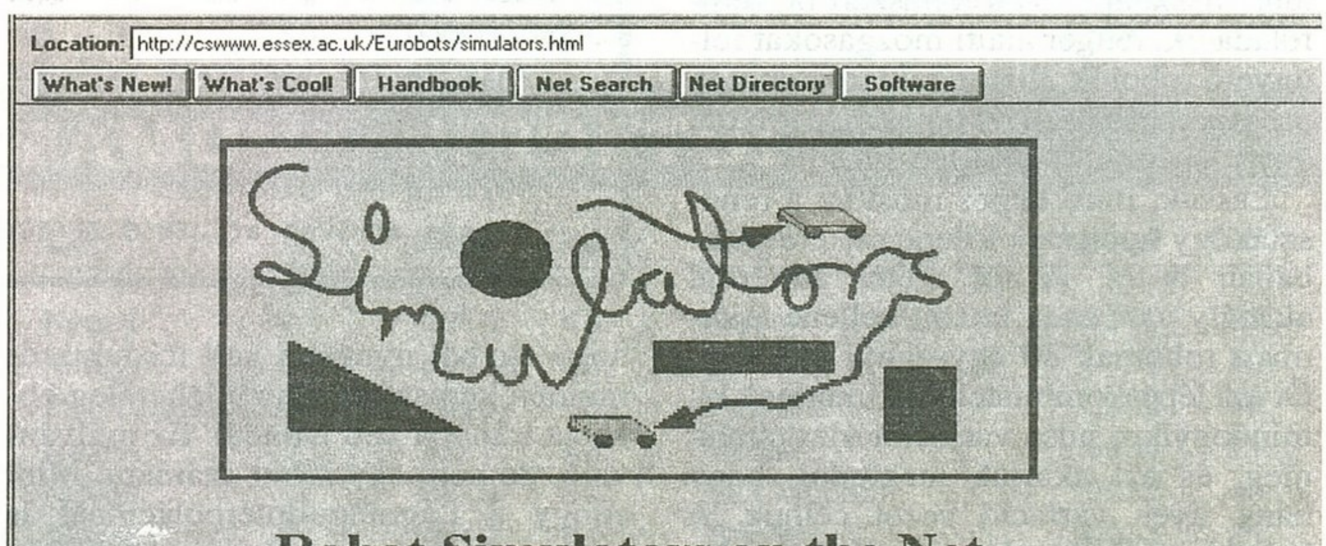
Nézzünk azonban egy egyszerű sétáló robotot. Igen gyorsan kiderül, hogy tud-e járni a gép vagy sem, azaz felborul-e vagy sem. S hogy ne boruljon föl, a robot nem tehet ki a képernyőjére egy homokórát, míg kiszámolja, hogy hova is fog most lépni, hogyan mozduljon, mert ha elkezdett dőlni, akkor azonnal cselekednie kell.

A legkülönbözőbb kísérletezgetésre használják a robotokat. James McLurkin (az MIT kutatója) egy hangyafarmot próbál létrehozni. Itt egy hangyának egy aprócska robot felel meg. Mindegyik robot egy halom érzékelővel van felruházva, így például fény-, infra-, ütközésérzékelőkkel. E hangyák infrafénnyel kommunikálhatnak egymással. Mielőtt képesek lennénk e hangyákat olyan viselkedéssel felruházni, amely már egy valódi hangyabolyra emlékeztetne, néhány egyszerűbb feladattal kell megpróbálkozni. Talán a legegyszerűbb ezek közül a „Mindenki kövesse a vezért!” feladat. Utána meg lehet tanítani a robotokat a fogócskára vagy a „Szerezd meg a zászlót” játékra. (Ha rajtam múlna, biztosan megpróbálnék egy labdajátékot is beléjük verni... Mire e cikk megjelenik, McLurkin valószínűleg lényeges eredményeket ért el. Erről az MIT WWW [http://www.ai.mit.edu/] szerverén biztosan találhatunk információt. Már most is vannak filmek a kész robotokról.)

Ugyancsak vannak filmek és fényképek M. B. Binnard lépegető robotjáról, Boardiceáról. Ez a robot még egy 45 fokos lejtőn is képes felmászni. A lépegető robotokat megfigyelve azt vesszük észre, hogy az ember nem a saját képmására alkotja ezeket a szerkezeteket. Ezen nem is lehet csodálkozni, hiszen már az is nagy dolog, hogy egyáltalán megállunk a két lábunkon, hiszen igencsak magasan van a súlypontunk. Ezért Binnard is a svábbogarat vizsgálgatja, és a tapasztaltak alapján tervezi robotjának lábait.

Társalkodás, társ-alkotás

Az átlagos felhasználó a tudományos-fantasztikus irodalomból ismert kommunikációt tartja elfogadha-


Location: http://cswww.essex.ac.uk/Eurobots/simulators.html

What's New! | What's Cool! | Handbook | Net Search | Net Directory | Software

Robot Simulators on the Net

This list is an an attempt to provide a centralized source of information on sites and groups that are producing **robot** simultion software which might be of interest to the academic or research community (eg freeware or shareware simulators sites). If you are new to this page, see comment on simulators. Wherever possible we are maintaining links to the original developers site (to ensure the latest copies!). However, where developers do not have their own Web or ftp servers we are happy to keep code locally at this site. Most Simulators are well described by associated ReadMe docs. This list is maintained by Vic Callaghan of Essex University and can only ever be as good as the information you provide, so please remember to keep him informed (callv@essex.ac.uk).

tónak, azaz élőszóban kíván társalogni a géppel. Ettől még messze állunk, de részeredmények már vannak. Például M.C. Torrance ugyancsak az MIT keretében foglalkozik azzal, hogyan kapcsoljon össze egy mozgó robotot egy természetes nyelvű be/ki menettel.

Mint a keretes anyag angol nyelvű példájából látható, a felhasználó bizonyos helyeket elnevezhet konkrétan vagy relatív kapcsolatokkal, és bonyolultabb feladatokat is megfogalmazhat (esetleg egyszerűbb feladatok sorozataként).

Nemcsak azt szeretnénk, ha szót értene velünk a gép, hanem ha egyedül is boldogulna a világban. Mi erre öt érzékszervet használunk, és ezek közül talán a legfontosabb a látás. Ha szétnézünk az állatvilágban, majdnem minden értelmesebb lénynek több szeme is van. Ez szinte elengedhetetlen ahhoz, hogy tájékozódjon a környezetéről, az őt körülvevő tárgyak távolságáról.

Ha modellezni szeretnénk a szemet, nem árt egy jó adag tehetetlenséget is beprogramozni a felismerés folyamatába, hogy ne csapja be a gépet egy villámgyorsan mozgó tárgy vagy az eső, esetleg a hóesés. Segíthetjük is a látást, mint ahogy a Carnegie Mellon egyetemen teszik. Ott egy radar elvégzi a távolságok mérését, így a látvány felismerése kicsit könnyebb. (Jó mulatság megtréfálni az így felkészített rendszert: ha a Vidám Parkból ismert tükörterembe tesszük, talán még jobban belezavarodik, mint mi.)

Samut (pontosabban SAMUEL-t, amely egy programrendszer) az összeütközések elkerülésére, illetve navigációs problémák megoldására kezdik használni. A későbbiekben talán fel lehet használni ezt a rendszert őrszemfeladatok, tenger alatti mozgásokat felügyelő robotok, illetve raktárosrobotok programozására is. (Meglepő, de Samu egy 486-oson, DOS alatt fut.)

Lássuk, mire képes most ez a rendszer: egy nyolcszor kilenc méteres szobában öt-tíz, jó fél méter átmérőjű akadály van, ezek között kellene átjutnia a robotnak 80 egységnyi idő alatt. Ötven lépéskombinációból indulunk ki, mindegyikre húsz variációt vizsgálunk meg, és azt akarjuk megtudni, hogy hány ilyen variáció vezet célhoz. A legjobb eredményeket adó lépéskombinációkat egy hosszabb menetben újfent teszteljük a variációkkal, és a kapott legjobb eredményt fogja végrehajtani a robot. Huszonnyolc különböző helyzetből 24 esetben idejében ért célhoz a robot.

Gyakori feladat: egy adott pontból el kell jutni egy másikba, mondjuk Debrecenből Pestre. Számunkra világos, hogy egyből nyugatra kell fordulni, és minden kereszteződésnél lehetőleg ebbe az irányba kell haladni. Ez nyilvánvaló, de nem egy robot számára. Mint ahogy a Lagrange-interpolációnál is jobb eredményt ad a Hermite (mivel itt nemcsak az értékeket ismerjük, hanem a keresett függvény meredekségét is), a robot is hamarabb célba ér, ha folyamatosan a cél irányába halad. (Ezt egy kicsit bonyolultabban és precízebben megfogalmazó cikket az amerikai légierő szponzorálta, valószínűleg nem csak passzióból.)

Igen hosszadalmas lenne, ha mindenre felkészítenénk a robotot. Ennél jobban járunk, ha kap egy minimális útbaigazítást, és mindaddig dolgozik, amíg tudja, miről van szó. Ha a felvetett problémával nem tud mit kezdeni, visszakérdez: ezt hogyan kell csinálni. Ekkor a megoldandó feladatokat további részekre bontjuk, s azokat a robot már képes megoldani. Miközben megoldja, meg is tanulja a megoldást, esetleg még visszakérdez, hogy pontosítson bizonyos részleteket. Próbálkozások vannak a mesterséges élet felhasználá-

Robottanítási példa

```
Where are you?
   I don't know where I am right now.
Which way are you facing?
   I don't know which way I am facing. Please tell me.
You are facing north.
   I am facing NORTH. Thanks.
You are at Mark's office.
   I'm learning about MARK'S OFFICE.
Turn around.
   I'm turning around.
Go until you can turn right.
   I'm going until I see no obstacle on the right.
You are at the northeast entrance to the elevator lobby.
   I'm learning about THE NORTHEAST ENTRANCE TO THE ELEVATOR LOBBY.
Go.
   I'm going.
Stop.
   I'm stopping.
You are at Jim's office
   I'm learning about JIM'S OFFICE.
How do you get from Jim's office to Mark's office?
   To get from JIM'S OFFICE to MARK'S OFFICE,
   In the ROP, I face NORTH and go 8.2 feet.
   In the ROP, I face NORTH and go 13.6 feet.
Karen's office is north of Jim's office.
   I'm learning about KAREN'S OFFICE.
Go to Mark's office
   Now I'm going from JIM'S OFFICE to THE NORTHEAST ENTRANCE TO THE
ELEVATOR LOBBY.
   Would you please tell me if I pass KAREN'S OFFICE.
   This is THE NORTHEAST ENTRANCE TO THE ELEVATOR LOBBY.
   Now I'm going from THE NORTHEAST ENTRANCE TO THE ELEVATOR LOBBY
to MARK'S OFFICE.
   This is MARK'S OFFICE.
   I have arrived at MARK'S OFFICE.
Go to Karen's office.
   I don't know how to get to KAREN'S OFFICE from here.
Go to Jim's office.
   Now I'm going from MARK'S OFFICE to THE NORTHEAST ENTRANCE TO
THE ELEVATOR LOBBY.
   This is THE NORTHEAST ENTRANCE TO THE ELEVATOR LOBBY.
   Now I'm going from THE NORTHEAST ENTRANCE TO THE ELEVATOR LOBBY
to JIM'S OFFICE.
   This is JIM'S OFFICE.
   I have arrived at JIM'S OFFICE.
```

sára is, azaz a tervezőtől alig függő rendszerek létrehozására is. (Lásd erről: Új Alaplap, 1996/7. szám.)

A lemezmelléklet apropóján

Az lenne az igazi, ha a lemezmellékletre egy olyan program kerülhetne, amellyel a laphoz mellékelt aprócska robotot irányítani lehet. Mivel ilyen hardvermelléklet egyelőre csak álmainkban létezik, önmagában a program sem sokat érne. A SimTel programbankban találhatunk viszont olyan programot, amely egy hatlábú robot mozgását mutatja be. Ezt mérete és hardverigénye miatt az őszre tervezett, 650 MB-os CD-ROM mellékletünkre igyekszünk majd rátenni. A SimTel programbázisban msdos/education/modulsh3.zip néven egy olyan program található, amellyel az általunk megtervezett egykarú robot végrehajtja programunkat. (Sajnos matematikai segédprocesszor nélkül el sem indul.)

Felrakhattuk volna még a lemezre a cs.cmu.edu gépen is megtalálható PCROBOT programot, ahol egy szobába bezárt robotok lövik halomra egymást. Nekünk kell megírnunk azonban a robot programját, viszont nincs megkötve a kezünk semmilyen programnyelvvel. Ráakadtam egy kicsit terjedelmesebb, de annál látványosabb programra (többek között ftp.essex.ac.uk/robot/simulators/NuTank címen található meg). Sajnos e program élvezetéhez is szükséges egy gyors és színes gép, bár szerényebb keretek között is kipróbálható.

Valaha a Tudomány című lapban a Számítógépes észjáték sorozatban szerepelt egy kis kocsikból és lámpákból álló világ leírása, ahol az autókban a fényérzékelők össze voltak kapcsolva a kerekekkel. E kapcsolatok milyenségétől függően egyes kocsik fényimádók lettek, mások messze elkerülték a fényt, néhányan meg nekirohantak a lámpának, így a fényimádókat máshova űzték. Noha a valóságban ilyen rendszert nem építettek fel, ám ezt nem nehéz dolog számítógépen szimulálni. A lemezmellékleten szereplő program e kocsiknál értelmesebb lényeket használ. Itt egy robotnak lehet szeme, orra, füle, gyomra (illetve ezeknek megfelelő érzékelők) és két kereke. Az már rajtunk múlik, hogyan kapcsoljuk össze ezeket, azaz ha érzi a kaja szagát, feléje fordul, avagy elfordul tőle.

Ismét a neuronhálóval...

Az élő szervezeteket modellezendő, az összekapcsolást neuronhálóval valósítjuk meg. Megadhatjuk az egyes idegpályák irányát és súlyát, de lehetnek önmaguk súlyát változtató idegek is, így nincs kizárva a tanuló robot sem. Ezután az így felkészített robotot kidobjuk az ÉLET-be, pontosabban egy mesterséges világba, ahol túl kell élnie társát. Ha nagyon jó kis robotot terveztünk, és a való életben is ki akarjuk próbálni, vegyük meg e program teljes változatát, mert abból már egy valódi robotba is betölthetjük ezt a programot.

A program szerzői néhány megszorítást építettek a programba. Ha nem szeretnénk nagyon sokat vesződni, akkor a C:\NUTANK alkönyvtárba telepítsük a programot. A program mellé adott cikket érdemes elolvasni, sok érdekességet tartalmaz. Végül egy ehhez hasonló, ám kicsit egyszerűbb program mellett döntöttünk. Itt egy svábbogár áll a középpontban, ennek mozgását irányíthatjuk. Maga a bogár itt is egy neuronhálóval összekapcsolt érzékelők és mozgatóberendezések összessége. A mozgás szintén energiát emészt fel, és ezt táplálkozással pótolhatja a svábbogár.

Majd kétszáz neuron tucatnyi paraméterét állíthatjuk be, és az így módosított bogár mozgását követhetjük akár lépésről lépésre is. Folyamatosan egyszerre több neuron elektronszintjét kísérhetjük figyelemmel. A lassabb gépek tulajdonosai örömmel fedezhetik fel azt a lehetőséget, hogy magára hagyhatjuk a gépet, amíg kiszámítja a svábbogár mozgását és rögzíti lemezen. (Ha mégis szeretnénk egy ilyen gépen nyomon követni az eseményeket, választhatunk nagyobb lépésközöket is.) Később természetesen lefuttathatjuk az így készült filmet.

Neuronok variálásával próbálkozók hasznosnak találhatják a neuronfájlokat olvasható formára konvertáló programot. A program mellett megtalálhatjuk a forráskódot is, TurboC formában, így egy kis munkával más operációs rendszer alá is lefordíthatjuk a programot, vagy más hasonló, egyszerűbb neuronhálót (vagy azt használó alkalmazást) készíthetünk.

Aki komolyabban elmélyülne ebben a programban és az ehhez kapcsolódó elméletben, haszonnal forgathatja dr. Beernek a dokumentációban ajánlott könyvét.

Az Alaplap-cikkek kapcsán már sokszor említett cs.cmu.edu szinte kimeríthetetlen forrása a robotikával foglalkozóknak. A comp.ai.robotics NetNews-csoport főleg azoknak érdekes, akiket hardverproblémák érdekelnek, és nagyjából csak ez szerepel a FAQ-ban is.

Aszalós László

A phaisztoszi korong vallatása

Közelebb a szöveghez

A legutóbbi alkalommal igen lényeges ponton sikerült előrehaladást elérnünk. A korong nyelvének leggyakoribb végződésében a görög nyelvnek egy ősi eredetű, archaikus névszói esetragját sikerült fellelnünk, amelyet eddig főleg a homéroszi nyelvből ismertünk.

Már Homérosz idejében is régiesnek érezhették az egyes birtokos eset -OJO végződését, de még elevenen élhetett. Ez abból valószínűsíthető, hogy az Iliászban és az Odüsszeiában ez az alak a vers ritmusának megfelelően váltakozik a később keletkezett -OU alakkal. (Először a J tűnt el, majd a második O átment U-ba. Egyébként Homérosznál az -OO változat is elég sok helyen megtalálható.) A korongon ennek a végződésnek a gyakori szereplése arra enged következtetni, hogy a korong készítése idején még teljes értékű hangként élt a „j" hang, amelyet a klasszikus görögben már inkább csak közvetve, hatásában tudunk kimutatni.

A továbblépéshez most az kellene, hogy a szókincs titkait is fel tudjuk tárni. Eddigi eredményeink alapján, amelyeket a szótagok olvasatának meghatározásában elértünk, a szövegbe való betöréshez legalkalmasabb helynek az A oldal első felében sokszor előforduló „szuperhosszú" szavak látszanak, mivel ezeknél sűrűsödnek legjobban a már meghatározott olvasatú szótagok. A szövegnek ez a része egyébként is különös figyelmet érdemel. Valószínűleg szemantikailag is itt juthatunk legközelebb a korong teljes szövegének fő mondanivalójához. Erre következtethetünk a fokozott mértékű szóismétlésből és ennek a szövegrésznek különlegesen erős ritmikáról árulkodó felépítéséből.

Ritmikai hullámzás

A szavak hosszának vizsgálata önmagában is alkalmas arra, hogy alátámassza e szövegrész ritmikájára vonatkozó megállapításunkat. Egyedül itt vannak az egész szövegben „szuperhosszú" szavak, s érdekes módon mindenütt „ultrarövid" szavak kíséretében. Ha soronként végigfutunk ezen a részen, a szavak hosszából is érződik, a szemmel látható alapján is szinte „hallhatóvá válik" a szöveg ritmikus hullámzása.

Mindjárt a bevezető sorokban felbukkan az első 7 szótagos, szuperhosszú szó, amely aztán a 6. sor elején újra visszatér. (Betűkhöz szokott gondolkodásunknak talán nem is tűnik a 7 szótag olyan soknak, de gondoljunk csak arra, hogy a Hódmezővásárhely vagy a Szentkirályszabadja még csak hat szótag.)

1. sor: 3(A1) + 2(A2) + 7(A3)
2. sor: 3(A4) + 3(A5) + 4(A6)

Csak emlékeztetőül: mindkét sor ugyanazzal a szóval kezdődik, és ugyanazzal a -TOJO toldalékkal végződik; az A6 szó töveként pedig az A2 szó ismétlődik meg.

A 4. sorral kezdődő három további sor ritmikai felépítése teljesen megegyezik: az elején mindegyiknél egy szuperhosszú szó áll (-TOJO végződéssel), a végén egy ultrarövid, s ezek mindenütt ugyanazt a 4 szótagos (-TOJO végződésű) szót fogják közre (1. ábra). A szóismétlés természetesen csak fokozza az ismétlődő ritmus által keltett hatást (A17 = A11; A16 = A13 = A10; A18 = A12).

4. sor: 7(A9) + 4(A10) + 2(A11)
5. sor: 6(A12) + 4(A13) + 2(A14)
6. sor: 7(A15) + 4(A16) + 2(A17),
7. sor: 6(A18)...

A szintaktikai szerkezet szemügyre vétele alapján újabb megállapítással egészíthetjük ki nyelvi megfigyeléseinket. Alaktanilag, mint láttuk, a -TOJO végződés névszók egyes számú birtokos esetét jelölheti. A szöveg szerkezetének megfigyeléséből most a birtokos eset használatának mondattani szerepét is kikövetkeztethetjük: feltehetően a legtöbb helyen névszói jelzőket kell bennük látnunk.

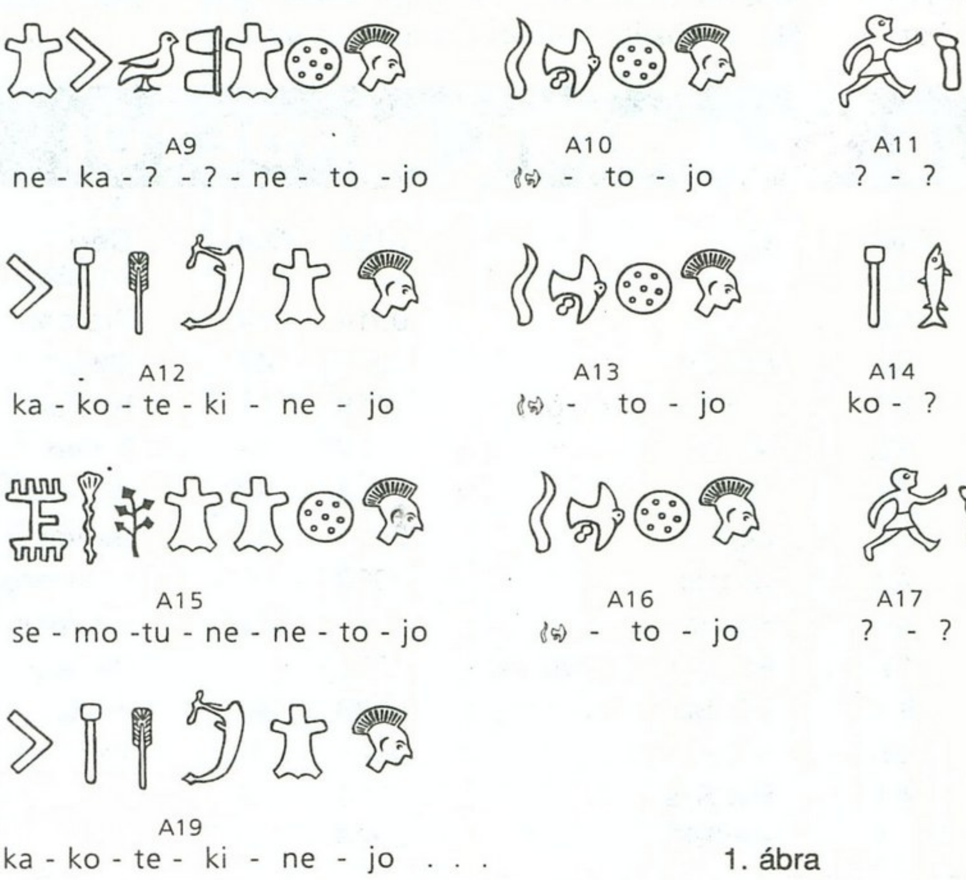
A lyukak kitöltése

Két olyan szuperhosszú szavunk is van, amelyek segítséget nyújthatnak a továbbhaladáshoz: az A3 (=A15) és az A12 (=A18) szó. Mindkettőből hiányzik ugyanannak a jelnek, az „irhá"-nak az olvasata. Az első szó mindjárt teljessé is válna, ha ennek a szótagjelnek az olvasatát sikerülne meghatároznunk.

A korong többi része alapján két fontos dolgot lehet tudni erről a szótagról:

1. Feltűnően gyakran fordul elő szavak végződéseként (A25, A26, B9, B19, B29), ami újabb névszói esetvégződésre enged következtetni. (Ez nem feltétlenül jelenti azt, hogy a magánhangzó is az esetvégződéshez tartozik. Gondoljunk csak a ciprusiakra, akik a mássalhangzóval végződő szavak írását úgy oldották meg, hogy egy néma e-vel toldották meg a szót. Például a görög ANTHROPOSZ [=ember] szó ciprusi szövegekben A - TO - RO - PO - SE alakban jelenik meg.)

2. A -TOJO vagy a -JO végződés előtt is több helyen megtalálható, éppen a szuperhosszú szavakban



1. ábra

(A3 = A15, A18 = A12). Szavakon belüli pozíciójából feltételezhető, hogy ilyenkor képzőt jelöl, de persze a tőhöz is tartozhat.

Irha és kampó

Az „irha" jelentésű szavak közül elsősorban két olyan szó került gyanúba, amelyekből az akrofóniai elv alapján megmagyarázható lenne ennek a szótagjelnek a keletkezése. (Szűrő feltételként természetesen azt az információt is érdemes felhasználnunk, amely az előbb említett pontokban rejlik.) Az egyik szó: NEBRIS vagy NEBRÉ, „szarvasborjúbőr". Ennek a szónak az az érdekessége, hogy szorosan kötődött Bakkhosznak a kultuszához: a bakkhánsok és bakkhánsnők ilyen állatbőrt viseltek Bakkhosz ünnepén. A másik szó: NAKOS vagy NAKÉ „gyapjas bőr, irha".

Szuperhosszú szavainkra alkalmazva a NE szótagjel alkalmasabbnak látszik. Ezt behelyettesítve az A3 (=A15) szó esetében a teljes olvasat SZE - MO - TU - NE - NE - TO - JO lenne. Az A12 (=A18) a teljes olvasatához előbb még meg kell határoznunk, hogy mi az első szótag: „? - KO - TE - KI - NE - JO".

Ha minden igaz, fejtőtársammal, Botos Imre nyelvész kollégámmal mindkét szó görög megfelelőjét sikerült megtalálnunk, és érdekes módon előbb a második szóval kapcsolatban jutottunk eredményre. Igen jól illeszthetőnek találtuk az A12 szót a görög KAKOTEKHNÉSZ, illetve KAKOTEKHNOSZ „ravasz, ármányos, csalárd, csábító, fondorlatos, hamis" szóhoz. (Igei megfelelője: KAKOTEKHNEO „ravaszkodik, mesterkedik, ármánykodik, elront, megront".) Ennek az összetett szónak az első összetevője egyébként a magyar fül számára sem idegen: a KAKOPHÓNIA, vagyis a rossz hangzás jelentésű szót ismerhetjük. Egyébként a görögben a KAKOSZ „rossz" szó számtalan összetett szó előtagja.

A „sarok" szótag KA olvasata is igen jól beleillik eddigi megfigyeléseink rendszerébe. Mint helyzete alapján előre meg lehetett jósolni, alliterál a lineáris B alapján kikövetkeztetett KO szótaggal. A KA szótag meghatározásával tökéletessé válik az első oldal végének és az elejének az egybecsengése: az A oldal utolsó két szava, az „A30 + A31" záróakkord hangtanilag szinte pontosan megfelel a bevezető sorok „A5 + A6" szópárjának (2. ábra). Az már csak hab a tortán, hogy a B oldal elején a B3 és B4 szavak alliterációja ezáltal két szótagos alliterációvá erősödik, és a második szótag belső alliterációként illeszkedik a B2 szó KIKO- kezdetéhez (3. ábra).

A KA olvasatú „sarok" szótag keletkezésének akrofóniai motivációja is megtalálható. Elsősorban a görög KAMPSZISZ „hajlat, görbület" szóra kell gondolnunk, amelynek litván megfelelője, a KAMPAS szó ma is „sarok", „szög" értelemben használatos.

Ismét Homérosz?

Nehezebb diót jelentett az A3 (=A15) szó megfejtése. Annyi az eddigiek alapján már valószínűnek látszott, hogy a SZE-MO-TU-NE-NE-TO-JO szónak is összetettnek kell lennie, de már az összetétel határainak megkeresése sem volt könnyű feladat. Különösen furcsa a NE szótagnak egymás utáni kétszeres előfordulása, főleg a szó közepén.

A megoldáshoz akkor jutottunk közelebb, amikor rájöttünk, hogy a szóhatár a szótagjel mássalhangzóját és magánhangzóját is szétvághatja. Valóban, ha az első NE szótag mássalhangzója még a szó első feléhez tartozik, akkor innentől kezdve érthetővé válik a szó második fele. Az ENETOJO alak az ENETOSZ „behatoló", „befelé igyekvő" szónak a birtokos esete lehet. De mi lehet akkor a SZEMOTUN?

A megoldáshoz az ötletet ismét az Iliász adta, amikor rábukkantunk a IV. énekben egy roppant különös, mint később kiderült, igen sokat magyarázott szópárosításra. Az ODUNÉ „fájdalom" szó nem tartozik a ritka szavak közé. Azon viszont már nagyon sokat vitatkoztak a Homérosz-kutatók, hogy mit is jelenthet pontosan a 117. sorban a HERM' ODUNAÓN kifejezés. Íme, néhány megoldás a magyar Iliász-fordításokból:

„Felnyitván tegezét, nyílvesszőt vett ki magának,
Szárnyasat és kilövetlent még, a sötét BAJOK ANYJÁT"... (Kempf, 1891)

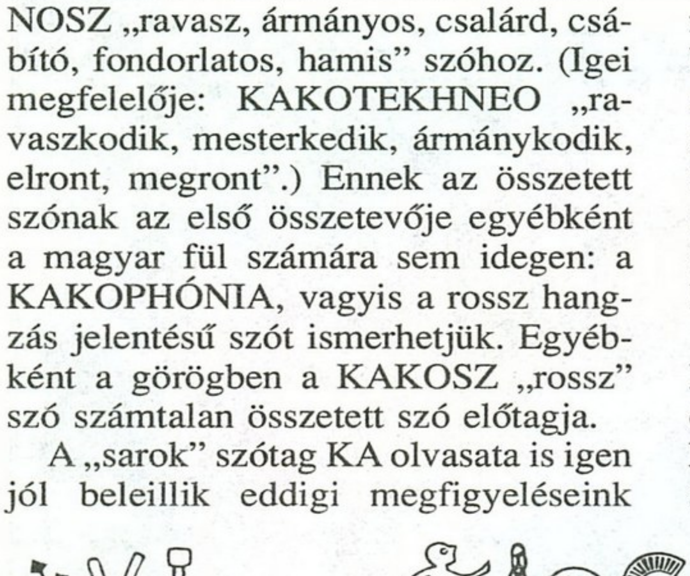
„Most megnyitja tegzét, választja vesszőjét,
Gyorsat, újat, KÍNOK sötét HÍRVIVŐJÉT"... (Baksay, 1901)

Csengery (1937) „KESERVES KÍNSZERZŐ"-ről ír, Devecseri (1974) „éjszínű FÁJDALMAK ÜTŐJÉ"-ről. A problémát tulajdonképpen a HERMA szó értelmezése jelenti. Legtöbben úgy fogják fel, hogy valaminek a forrásáról, okozójáról van szó. De a „fájdalmak HERMÁJÁ"-ról asszociálhattak a görögök a csalafinta Hermészre is, aki később az istenek hírvivője lett, hiszen a furfangos isten tiszteletére felállított útszéli oszlopokat HERMÁ-knak hívták. Kerényi Károly ezt írja Hermész kultikus ábrázolásával kapcsolatban: „Egy fából vagy kőből való phallos jelölte Hermést, vagy ezzel rokon alakban: egy nyelvünkön hermának nevezett négyszögletes kőoszlop, aminek feje volt és fölálló phallosa"...

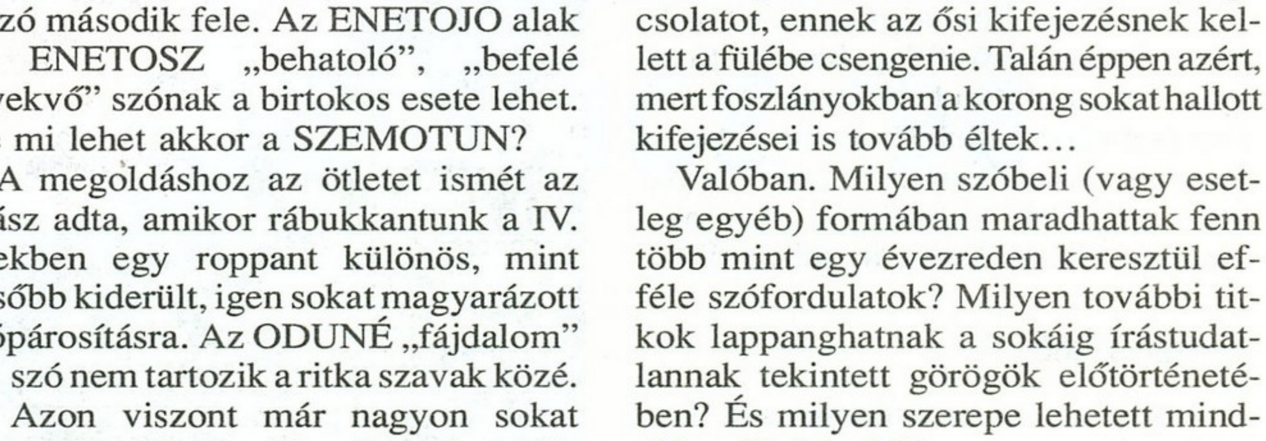
Hogy minden félremagyarázási lehetőséget elkerüljünk, megelégedhetünk annyival, hogy egyszerűen a „fájdalmas" jelzővel fordítjuk a „SZEMOTUN" szóösszetételt. Ha ez az értelmezés helyes, akkor a korong SZERM' ODUN' szókapcsolata megkövesedett fordulatként kellett, hogy éljen Homérosznál a sokkal régebbi idők görögjeinek nyelvéből. Amikor Homérosz a hősök húsába behatoló nyíl jelzőjeként használja a „Herm' odunaón" szókapcsolatot, ennek az ősi kifejezésnek kellett a fülébe csengenie. Talán éppen azért, mert foszlányokban a korong sokat hallott kifejezései is tovább éltek...

Valóban. Milyen szóbeli (vagy esetleg egyéb) formában maradhattak fenn több mint egy évezreden keresztül efféle szófordulatok? Milyen további titkok lappanghatnak a sokáig írástudatlannak tekintett görögök előtörténetében? És milyen szerepe lehetett mindebben Krétának?

Vargha Dénes



2. ábra



3. ábra

Harc az élet...

A leveles játékok varázsa

Az ember szellemi frissességének megtartásához agyának rendszeres tornáztatása is elengedhetetlen. Ne csak a különféle „agysportokra" gondoljunk, hanem a játékokra is, így a sakkra, vagy kártyára és a társasjátékok népes családjára. Választhatjuk ellenfélnek a számítógépet is, de egy idő után unalmassá válik ugyanazt a programot használni, az meg, hogy rendszeres időközönként lecseréljük játékprogramjainkat, felettébb költséges mulatság (lopni pedig ugye nem illik, és még drágább mulatság is lehet). Igen régi hagyománya van viszont a levélben játszható játékoknak....

A leveles játéknak megvan az a előnye, hogy nem kell összegyűlniük egy adott helyre és adott időben a játékosoknak, hanem tetszőleges időben és helyen fogalmazhatják meg következő lépésüket, azt pedig levélben elküldhetik a játékmesternek. A játékmester — miután összegyűjtötte a leveleket, megteszi a felírt lépéseket, és a végeredményről tájékoztatja a játszó feleket. (Legalábbis azon részekről, amelyek az adott játékosra vonatkoznak.) Tehát ha épp valamilyen kártyajátékot játszunk, nem fogjuk megtudni idő előtt, hogy az ellenfeleknek milyen lapjai vannak.

A játékmesternek a postaköltségen felül nem kis fáradságába kerül valamennyi résztvevő (főleg, ha több tucatnyian vannak) minden lépésének követése és a válaszlevelek megírása. Lehet, hogy ez szórakoztató, de azért elenyésző kivételtől eltekintve bizonyos összeget elkérnek érte a játékosoktól.

„Megáll az idő..."

Angliában és Amerikában sok ilyen játékot játszanak, nálunk főleg a levélsakk terjedt el (ehhez viszont nem kell játékmester), és az utóbbi időben megjelent egy-két szerepjátszó levéljáték is. Ha nincsenek fizikailag közel a játékosok, igen sokat kell várni a postafordultára, és a levélköltségek is magasak.

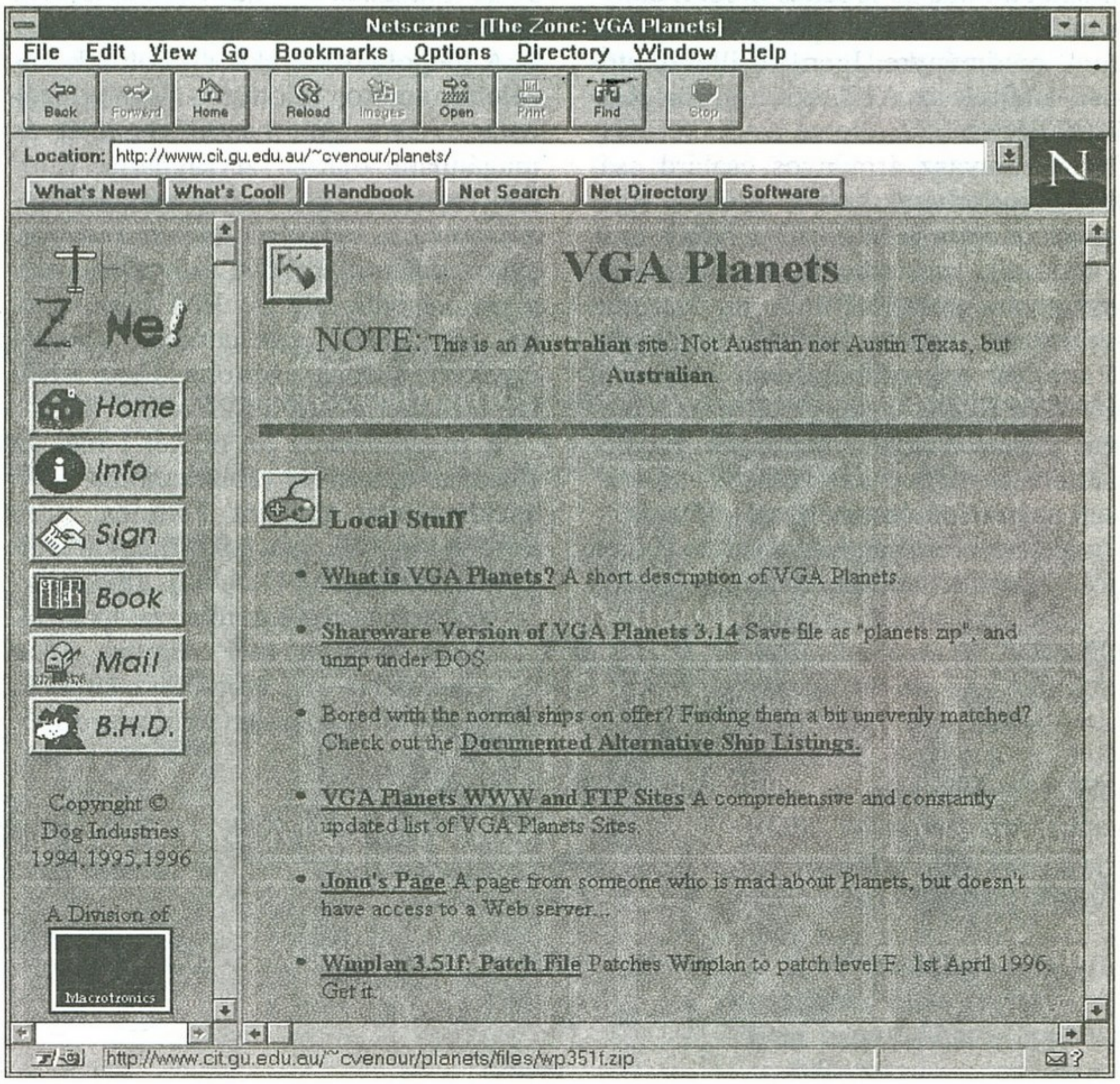
Valaha egy testvérpár igen nagy összegben fogadott egy köztük zajló sakkjátszmára. Az egyik Európában, a másik az óceánon túl lakott, és a posta szerepét betöltő vitorlások nem voltak igazán gyorsak. Ezért eshetett meg, hogy fél évig kellett várni a válaszlépésre, és amikor az egyik testvér már majdnem megnyerte a játszmát, meghalt. Noha az örökösök a nyilvánvaló győzelem tudatában perelték a másik testvért, a bíróság ítélete szerint (ami talán még most is érvényes) az egyik játékos halála esetén a játék döntetlenül ér véget. Mióta a posta felgyorsult, ilyenre már nemigen kell számítani, de a játékok még mindig lassúak.

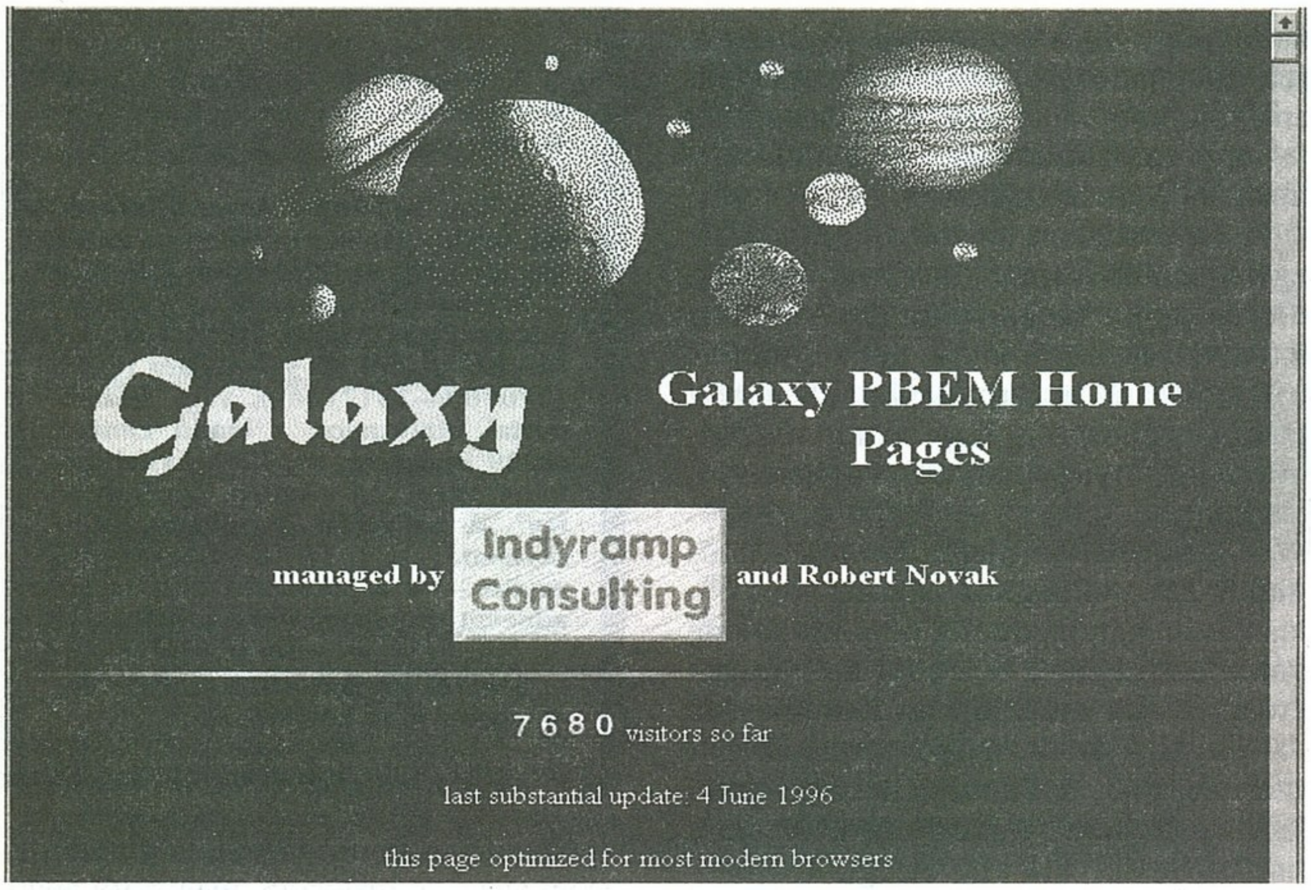
Háló(zato)san...

Ezeket a hátrányokat ki lehet küszöbölni az e-mail használatával. Egyrészt pörög a játék, másrészt csökkenti a költségeket (és azt sem a postának kell fizetni), másrészt megjelentek a gépi játékmesterek. Ha a játékosoktól megköveteljük, hogy előre meghatározott formátumú levelet írjanak, akkor azt egyszerű lesz számítógéppel feldolgozni, és megalkothatunk egy olyan rendszert, amely minden emberi beavatkozás nélkül fogadja és feldolgozza a leveleket, illetve postázza a választ.

Természetesen vannak olyan játékok, amelyekben elengedhetetlen, hogy egy élő személy tartsa kézben az események szálait és bogozza ki azokat legjobb tudása szerint (ilyen típusúak például a szerepjátszók), de őt is segítheti az előzőhöz hasonló rendszer.

Az Interneten elérhető, akár ember, akár gép vezette játékok között egyaránt található ingyenes és fizetős. E játékok legjobb tudomásom szerint angol és (kisebb mértékben) német nyel-





vűek. Van köztük olyan, amelyhez minimális nyelvtudás is elegendő, és akad olyan is, ahol tárgyalnunk kell a játékostársakkal. (Kíváncsi vagyok, lesz-e hazánkban valakiben annyi spiritusz, hogy magyar nyelvű és esetleg magyar témájú játékot írjon, illetve vezessen?) Ha nem építünk be a játékba gépi irányítású ellenfeleket, akkor ez egyszerű szabályok ellenőrzéséből, bizonyos paraméterek újraszámításából és az input/output műveletekből áll. Tehát kellő irányítással akár egy kisiskolás is tudna ilyen programot írni!

Vannak olyan játékok, amelyekben a véletlennek is van szerepe, mint például a „Ki nevet a végén?" elnevezésű társasjátékban. A véletlent leggyakrabban kockadobálással érjük el. Ha olyan játékot játszunk az Internet segítségével, amelyben kockákat kell dobálni, felmerül a kérdés, hogy ki dobja fel a kockákat? (Minden szentnek maga felé gurul a kockája!) Az eldönthetetlennek tűnő problémán az Internet segít, mert ha a dice@danpost4.uni-c.dk címre elküldött levélben felsoroljuk kockadobási igényeinket, ez a gép az általunk megadott személyeknek (azaz nekem és az ellenfeleimnek) elküldi a kockadobások eredményeit. Bővebben erről a lemezmellékleten lehet olvasni.

Ha egyszer egy üzlet beindul!

Az ilyen levelezési játékokhoz kapcsolódó ipar is terjeszkedik. Több magazin teljes terjedelemben ilyen játékokkal foglalkozik, az erre szakosodott üzletekben játszókészletek és szabálygyűjtemények kaphatók. (Majdnem minden utcai könyvárusnál lehetett kapni a Mágust, az első magyar nyelvű AD&D szabálykönyvet.) Léteznek ingyenes magazinok is, ezek egyike a lemezmellékleten megtalálható. Mivel ezek csupán hobbiból készülnek, terjedelemben és minőségben jóval elmaradnak pénzes társaiktól, de ha ló nincs, jó a szamár is. Ez a példány eligazít az ingyenes játékok között, felsorolja, hogy hol kaphatunk róluk további információkat.

A lemezmellékleten szerepel egy majdnem húsz játékot ismerő játékszerver leírása, ahol más játékosok vagy gép ellen is játszhatunk. Itt főleg táblás játékok találhatók.

A játékok másik elterjedt csoportját a harci játékok alkotják. E játékok közül a számomra legjobban tetszőt mutatom be röviden. A neve a HT-n játszott lövöldözős játékot juttatja eszünkbe: Galaxy. A helyszín továbbra is az űr, de nem az ügyességre épít. Az egyszerűség kedvéért kétdimenziós csillagrendszer egyik bolygójáról indulunk, és célunk meghódítani az egészet. Természetesen a többi játékosnak is hasonló céljai vannak.

Galaxisokon át...

A bolygók között űrhajókkal közlekedhetünk, ám ezeknek az űrhajóknak a megtervezése ránk hárul, pontosabban a páncélzat, a fegyverzet, a hajtómű és a raktér méreteit kell megadnunk. Az adatokat megfelelően kombinálva készíthetünk bolygó felett keringő űrerődöket, pici futárgépeket, vagy akár megépíthetjük a StarWars-ból ismert halálcsillagot is. Rövid idő alatt többezres űrflottával rendelkezhetünk, amelynek egységeit külön-külön képtelenség mozgatni. Ezért azonos típusú űrhajókból csoportokat készíthetünk, amelyek együtt mozognak, míg másként nem határozunk (azaz feloszlatjuk vagy újraszervezzük a csoportot). A csoportokból flottát szervezhetünk, ám a flotta csak olyan gyorsan tud haladni, mint a leglassabb gépe.

Egy űrhajó sebességét a hajtómű és az össztömeg aránya, illetve az határozza meg, hogy milyen fejlett a hajtómű. Az előbbi azt jelenti, hogy egy teherszállító lassabban halad megrakodva, mint üresen. Szóljunk röviden a fejlettségről is: minden bolygónak van bizonyos ipari szintje és valamekkora la-

kossága. Főleg az ipari szint határozza meg (bár a lakosság száma is beleszól), hogy mennyit termel a bolygó egy fordulóban. A termelés egyik (igen költséges) formája a technológia fejlesztése, ennek révén gyorsabb űrhajókat, erősebb fegyvereket, jobb páncélokat és bővebb raktereket készíthetünk, tehát érdemes erre áldozni. Ha nem fejlesztésre költünk, bányászhatunk, űrhajót építhetünk vagy javíthatjuk az ipari szintet is. A népesség folyamatosan nő, és ha túllépi a bolygó méreteinek tűrőképességét, akkor a lakosok automatikusan leendő telepesekké minősülnek át, akikkel betelepíthetjük az üres bolygókat.

Minden bolygó monokultúrás, azaz ritka kivételtől eltekintve egyszerre csupán egy dolgot termelhet, azaz bűn egy ásványi anyagokban bővelkedő bolygón nem bányászni, ezért a bányaművelésre kevésbé alkalmas bolygókon kell űrhajót építeni. Ehhez természetesen ide kell szállítani a nyersanyagot. Ezt a szállítást bizonyos mértékig automatizálhatjuk is.

Naná, hogy harcolni is kell!

Miután ellenséges bolygókat is el kell majd foglalni, szükséges a harc, de ez csak a bolygók közelében lehetséges. Ha itt két ellenséges flotta találkozik, véletlenszerű lövéssorozatok során az egyik pusztulásával ér véget e küzdelem. Ha az ellenség bolygója felett győzünk, kellő sterilizálás után birtokba vehetjük a bolygót. Kezdetben mindenkivel harcban állunk, ám ezen változtathatunk. Egyoldalúan békét, illetve semlegességet jelenthetünk ki más államok ellen; kérdés, hogy ezt más államok hogyan fogadják.

Egyes verziókban lehetőség van államok közti üzenetváltásra, diplomáciára is. (Ez a legveszélyesebb fegyver!) Ha van még olyan állam, amellyel harcban állunk (és miért ne lenne), az átlátható hiperűr miatt idejében értesülhetünk róla, hogy mekkora össztömegű flotta közelít felénk, és mikor fog megérkezni. Azt csak későn tudjuk meg, hogy milyen űrhajók alkotják e flottát, és mennyi van belőlük, pedig gyakran ezen múlna a védekezési stratégia. Utolsó mentsvár az itt tanyázó ócska űrhajók modernizálása, és noha ez hatalmas költséggel jár, talán életben maradunk.

Ha egy űrhajó belépett a hiperűrbe, akkor onnan már nincs visszaút, még akkor sem, ha menet közben derül ki, hogy fejjel rohanunk a falnak. Az ellenfelekről csak annyit tudunk, amennyit látunk, így birodalmak emelkedhetnek fel és tűnhetnek el a süllyesztőben anélkül, hogy tudnánk róla valamit is — épp ezért küldjünk mindenfelé láthatatlan kémszondákat!

A lépéseket és a viszontválaszokat egy speciális formájú levélben küldjük, illetve kapjuk meg. A parancsok leírása, a válasz szerkezete és néhány leírás megtalálható a http://www.iquest.net/galaxy webcímen. Egy szótár segítségével már nyugodtan nekivághatunk a játéknak.

A játék méltán megérdemli a stratégiai jelzőt. A közgazdasági akciók, fejlesztések, harcok mellett jókora szerepe van a politikának is. Tudni kell, mikor érdemes szövetkezni más hatalmakkal — és kikkel —, hogy térdre kényszeríthessük az ellenfeleket, de tudni kell azt is, hogy mikor kell kiugrani a már terhessé váló szövetségből.

Van másik!

A Galaxy nem köti ki, milyen géptípuson levelezünk. Viszont csak VGA monitorral rendelkező PC-n játszhatjuk a VGA-Planet játékot. Ez témájában nagyon hasonlít a Galaxy-hoz, itt viszont nem begépelni kell a parancsokat, hanem menürendszerrel felszerelt programban kell kiválogatni lépéseinket, majd a program által ezek alapján elkészített fájlt (esetleges kódolás után) el kell küldeni a játékmesternek. Egy ilyen fájl mérete a 2-3 kbájt körül van, szemben a Galaxy-ban elérhető 100 kbájtos levéllel.

A VGA-Planet lépéseit és az arra adott válaszokat még távolsági hívással is olcsón továbbíthatjuk. Nem kell feltétlenül e-mail lehetőséggel rendelkezni, nyugaton nagyon sokan ezt a játékot BBS-en játsszák, ahol a BBS tulajdonosa könnyedén automatizálhatja játékmesteri feladatát. Számára a program ingyenes, míg az egyes játékosok programjai közprogramok, és ezeket egy idő után illik regisztráltatni (és hasznos is, mivel a regisztrált változatú programmal erősebb lesz a játékos).

A szükséges programokat megtalálhatjuk például az ftp://ftp.fu-berlin.de/pub/pc/msdos/games/vgaplanets címen, és még sok más helyen is. Semmi akadálya sincs, hogy nálunk is elterjedt legyen ez a játék, amely már évek óta megszakítás nélkül ott szerepel a száz legjobb számítógépes játék között. Az se baj, ha hálózat sincs, egy gépen is lehet játszani, csupán a megfelelő titkosságot kell megszervezni.

Jó játékot mindenkinek!

Aszalós László

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint.
A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11701004–20171649), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 157-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es mellék.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Megvételre vagy programokért cserébe **keresek** Logitech ScanMan 32-es vagy 256-os kéziszkennert. Ajánlatokat a 132-7243-as telefonon várok.

Főiskolai diák **oktatást, korrepetálást vállal számítástechnikából** kezdőknek, haladóknak, felvételire, pótvizsgára készülőknek. Cím: Kovács Gábor, 3502 Miskolc II., Pf. 83.

Computer Club! Írj, ha érdekelnek a szabad terjesztésű programok, játék demók! Küldd el számítógéped leírását! A belépés ingyenes, 0-99 éveseknek. Cím: Computer Club, Papp Ferenc, 4274 Hosszúpályi, Damjanich u. 8.

Vállalkozóknak teljeskörű **könyvelést vállalok** kedvezményes áron. Telefon: Svantek Andrea (27)314-897.

Szóra érdemes

Pétery Kristóf:

Word 7.0 Windows 95 alatt

Kézikönyv a program
magyar nyelvű változatához
A Mikroelektronika Alkalmazásának
Kultúrájáért Alapítvány
LSI Oktatóközpont, 1996
428 oldal, 1650 Ft

A Win95 megjelenése után nem váratott sokat a WinWord fejlesztőgárdája sem, hogy színre lépjen az új fejlesztéssel. Az új, igényesebb operációs rendszer akkor ér valamit, ha elegendő számban vannak hozzá szoftverek is, amelyek igénylik szolgáltatásait. Az pedig már csak természetes, hogy a Microsoftnak kell élen járnia fejlesztésben. Hadd érezze a felhasználó, hadd igazolhassa sajátmaga előtt is, hogy érdemes volt kiadnia a tömérdek pénzt rendszerének továbbfejlesztésére.

Kétségtelen, hogy a Word a cég legsikeresebb termékei közé tartozik. Igényessége, sokoldalúsága számos területen sikerrel kamatoztatható, az irodai munkától akár komolyabb kiadványok, egyedi brosúrák, sőt könyvek elkészítéséig. Nagyobb méretű, bonyolultabb szerkezetű anyagok kéziratának kidolgozásakor mindig a felhasználó fölött lebeg ugyan a lángpallos: ha nem darabolod kisebb fejezetekre a könyved, bármikor lesújthatok rád. Annyit látsz csak a világból, hogy „Nincs elég memória, zárj be valamilyen alkalmazást", és örülhetsz, ha némi ügyességgel ki tudod menekíteni keserves munkád újabb féligkész darabkáját egy floppyra... Aztán kezdheted újból a küzdelmet, jó varázslók és rossz boszorkányok társaságában, majd kiderül, hogy ki az erősebb: a feneketlen étvágyú óriás-e, vagy te, a felhasználó.

Számomra a Word ennek a lidérces üzenetnek az ismételgetésével bizonygatta állandóan, hogy ő érzi magát erősebbnek — a végén mégis én győztem. Igaz, az még a 6.0-s változat volt. Azóta bizonyára megtanulta, hogy a nem használt memóriaszeleteket néha illik elengedni, nem kell mindig újabbak után ácsingózni.

Pétery Kristóf, a könyv szerzője elég óvatos volt hozzá, hogy 16 Mbájtnyi RAM és gigabájtnyi méretű winchester nélkül ne is fogjon bele a tesztelésbe. Így viszont nem panaszkodik, hogy bármikor is kifogyott volna a memóriából. Pedig a Win95 sem nagyon türtőzteti magát, amíg van kitől elfoglalni újabb területeket.

A könyv olvasásakor — érthető módon — állandóan azt figyeltem, hogy mennyivel tud többet az új változat, mint az elődje. Legtöbbször bizony azt tapasztaltam, hogy inkább csak újdonsült gazdájának, a nagy Win95-nek akar a kedvében járni, és csak később találtam olyan dolgokat, amelyek más tekintetben is indokolják az új változat megjelentetését.

Pétery elsősorban a következő újdonságokat hangsúlyozza:

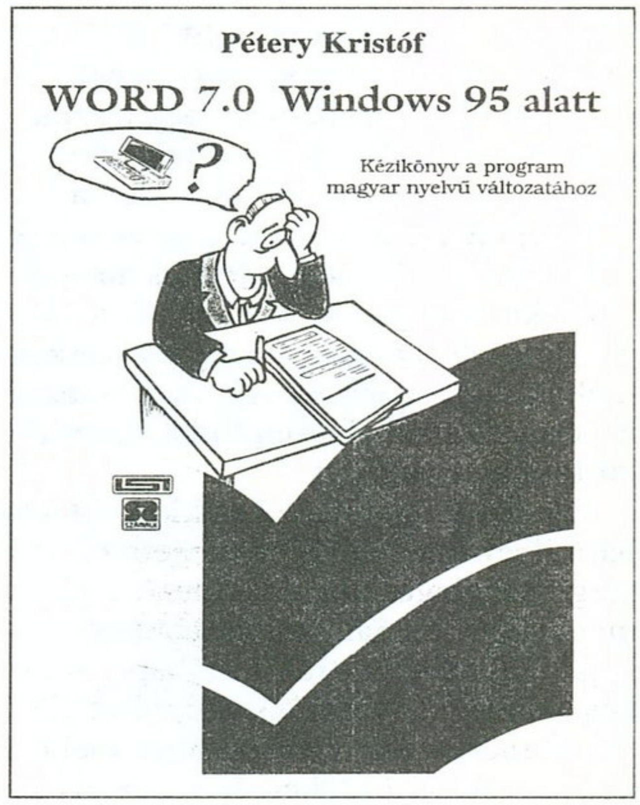
— Legnagyobb újítása a 7.0-nak a szerkesztés közben végzett, „online" helyesírás-ellenőrzés.

— A Word 7.0 a korábbi változatnál jóval több, 10 előre elkészített varázslót és 27 sablont szolgáltat a gyors és kényelmes munkához.

— A LevélVarázsló újdonsága, hogy az egyik párbeszédpanelen megjelenik a Címjegyzék nyomógomb, amellyel címjegyzékünket bővíthetjük, vagy onnan adatokat vehetünk át.

— Ugyanez az elektronikus postázás során is alkalmazható.

— Megjelent a szövegkiemelés: a kívánt szövegrészeket figyelemfelkeltés céljából feltűnő színnel festhetjük át.



— Az állapotjelzők nemcsak jelzik az aktuális üzemmódot, hanem be és ki is tudják kapcsolni.

— Az állománykezelésben jelentős változások történtek, bár az alapszolgáltatások nem változtak.

— Szerkeszteni is lehet nyomtatási kép üzemmódban.

Bármennyire jelentősek is ezek a változások, nem biztos, hogy emiatt érdemes a WinWord eddigi felhasználóinak rendszert és szövegszerkesztőt váltaniuk. A legfőbb aduként emlegetett „helyesírás-ellenőrző" életében valóban jelentős esemény, hogy már menet közben meg tudja jelölni a gyanúsnak észlelt szavakat, ám ennek a WinWordbe csak bedrótozott hazai programtermékneknek a fejlődése elég független attól, hogy mikor mi a befogadó rendszer. Arról sem lelkendezik a szerző, hogy bele volna építve a Word 7.0-ba olyan okos fejlesztés, amely végre kezdi kiérdemelni a helyesírás-ellenőrző nevet, mert a szintaktikai hibák egy részét is fel tudja deríteni.

Lényeges változást az jelenthetne a Word fejlődésében, ha elindulna a többnyelvű szövegek integrálása útján. Lehetetlen helyzet, hogy a magyar nyelvi

eszközök csak az angol HELYETT kerülhessenek bele a rendszerbe, s például az angol spelling checker vagy a szinonimaszótár kedvéért rendszert kelljen váltani (és vásárolni, persze). Ugyanígy érthetetlen korlátja a rendszernek, hogy nem lehet például egy előadás anyagában ékezethelyesen hivatkozni a magyar, a cseh, a román, a lengyel szerzők mellett a franciákra vagy az olaszokra is. Vagy ékezethelyesen írni a különböző földrajzi neveket.

Ha szimbólumokat be lehet ültetni bármilyen szövegbe, miért nem oldják meg tetszőleges ékezetes karakterek és módosított betűalakok beültetését — legalább néhány frekventáltabb fontkészletben? A Windings nagyrészt idétlen ábrácskáinak elérhetősége mellett ilyesmivel (is) törődhetnének a rendszer fejlesztői!

Mindez persze nem a könyv szerzőjének szól, de már csak a hónap témájára való tekintettel is indokoltnak láttuk megemlíteni. Lehetne kritizálni azt is, hogy a honosítás miért nem terjedt ki magyar levelezési sablonok kifejlesztésére. Emellett lefordítás nélkül lehetne megtartani az idegen nyelvű sablonokat is (lehetőleg persze ízléseseket válogatva) azok számára, akik külföldi partnerekkel is leveleznek. Persze ezzel megint visszakanyarodunk az előbbi témához: mi lesz az idegen karakterekkel? (Vagy az üzleti partnerek is csak a szomszédból kerülhetnek ki?)

Pétery igyekszik alapos munkát végezni a szoftver lehetőségeinek bemutatásában, s egészében valóban elmondható a könyvről, hogy gondos, precíz munka eredménye. Lelkiismeretesen igyekszik bemutatni a szoftver környezetéből mindazt, ami a kezdők számára nélkülözhetetlen, de mások számára is hasznos. Az Excelről írott könyvében alkalmazta először az ikonok tárgymutatóját, és itt ismét alkalmazza ezt az ügyes újítást.

Világosabban lehetne viszont az egész anyagot áttekinteni, ha szembetűnőbb lenne a tagolása, gazdagabb a kereszthivatkozások rendszere. Érdemes lett volna például élesen szétválasztani a kezdőknek és a haladóknak szánt részt a negyedik fejezet kezdeténél, viszont sűrűbb hivatkozással kötni össze a két részt. Ez bizonyára csökkentené az első rész olvasása közbeni hiányérzetet, és segítené a könyvben való tájékozódást.

Valószínűleg nem csak én találom például zavarónak, hogy sem az első előfordulás helyén (a 44. oldalon) nem derül ki, sem a tárgymutatóból, hogy a mezőkódokról és felhasználásukról bővebb információ is található a könyvben, jóval később.

Bizonyos mértékig a szerkesztés körébe tartozik az is, hogy olyan nagy betéteket, mint a mezőutasítások ismertetése, jobb lett volna a függelékben elhelyezni — különösen ilyen kategorizálás nélküli, ábécérendes felsorolásban. Ha viszont átkerült volna oda, akkor a szövegben lehetne egy rövid tematikus, áttekintő rendszerezés erről a hasznos és jól kidolgozott ismeretanyagról.

Tetszett a könyvben, ahogy a szerző összetettebb példán, a „Műszaki Irányelvek" elemzésén keresztül mutatja be a formázás gyakorlati megvalósítását. Érdemes lett volna hasonló módon bemutatni azt is — akár saját könyvrészletének példáján —, hogy az automatikusan létrehozható jegyzékek (tárgymutató, tartalomjegyzék, ábrák, egyenletek jegyzéke) hogyan készülnek.

Végül még egy enyhe szemrehányás: előfordul, hogy előzetesen tett ígéretei beváltatlanok maradnak. Bevezetőjében a részletes tárgymutató igénybevételére buzdítja az olvasót, az elkészült tárgymutató azonban rendkívül foghíjas. De saját újításával, az ikonok tárgymutatójával is hasonló a helyzet. Az előszó még teljességet ígér, mégis kimaradtak az állománykezeléssel kapcsolatos ikonok.

V. Nagy Edit

Keresse könyveinket!

COMPUTERBOOKS

Benkő T.-né-Benkő L.-Tóth B.-Varga B:
Programozzunk Turbo Pascal nyelven!
- bővített, átdolgozott kiadás
- 3.5-es lemezmelléklettel 1.490.-

Dr.Kovácsné C.J.-Ozsváth M.-G.Nagy J.:
OFFICE 95 - Irodai alkalmazások példákkal, feladatokkal 1.568.-

dr.Tamás P-Tóth B. és trai:
WINDOWS 95 + Microsoft PLUS! felhasználóknak - magyar nyelvű változathoz 1.960.-

Tóth B.-dr.Tamás P. és trsai:
WINDOWS 95 & Microsoft PLUS felhasználóknak - angol nyelvű változathoz 1.995.-

Jakab-Juhász-Vémi: **Adobe Photoshop** 2.480.-

dr.Tamás P.-Tóth B.-és trsai: **DELPHI - út a jövőbe** - lemezmelléklettel 1.999.-

Kérje katalógusunkat!

Levélcím:
1253 Budapest Pf.: 71.
Bp.,XII.Tartsay V.u.12.
Tel.: 175-1564
Tel./fax: 175-3591

INFORMÁCIÓKÉRÉS: 0906 ▼

PROFON INFORMATIKAI Kft.

AMI MINDIG MINDENT ÖSSZEKÖT

SZÁMÍTÁSTECHNIKAI RENDSZEREK KOMPLETT HÁLÓZATÁNAK TERVEZÉSE ÉS KIVITELEZÉSE

- **ADATHÁLÓZATOK**
  UTP, IBM Cabling System; ETHRNET, TWINAXIÁL, OPTIKAI hálózatok
- **ERŐSÁRAMÚ HÁLÓZATOK**
  Számítástechnikai rendszerek főhálózattól független speciális energiaellátása
- **HÍRKÖZLŐ HÁLÓZATOK**
  Alközponti, modemes, fővonali hálózatok
  Hírközlő és számítástechnikai hálózatok egy nyomvonalon
- **EGYEDI NYOMVONALAK KIÉPÍTÉSE**
  Egyedi facsatorna, faparapet tervezése és kivitelezése
- **ÖSSZEKÖTŐ KÁBELEK**
- **RACKSZEKRÉNYEK, RACKSZERELVÉNYEK**
- **RiT HÁLÓZATÉPÍTŐ ELEMEK**
- **PROFON, ÉS PACKARD BELL SZÁMÍTÓGÉPEK**

Rövid kivitelezési határidő, hároméves garancia!

1138 Budapest Cserhalom út 4.
Telefon: 270-6227, 270-6235
Telefon/Fax: 270-50-93

MASTER'S Grafika

INFORMÁCIÓKÉRÉS: 0927 ▼

Mia

VARESZI KFT.
1153 Budapest XV., Deák Ferenc u. 3.
Tel./Fax: 169-4206

ÚJ!
SZÍNVONALAS!
HATÉKONY!
REKLÁM ÉS HIRDETÉS

A VARESZI KFT.
a **számítógépes multimédia** eszközeivel kínálja fel, hogy az

ÖN cégét, termékeit és szolgáltatásait

Kelet-Európa legnagyobb új üzletközpontjában,
az **1996. novemberi** átadásra meghirdetett
PÓLUS CENTER
Bevásárló és Szórakoztató Központban

folyamatosan hirdeti és reklámozza.

Az új, színvonalas reklámlehetőség **napi 30–40 000 érdeklődő, vásárló** előtt biztosítja a folyamatos piaci jelenlétet.

A VARESZI KFT. által nyújtott szolgáltatás igénybevétele esetén az Ön lehetőségeit a CD lemezen kiadásra kerülő M I A adatbázisban térítésmentesen is megjelentetjük.

INFORMÁCIÓKÉRÉS: 0936 ▼

SZÁMÍTÁSTECHNIKAI TANFOLYAMOK AZ OGIMEX SZÖVETKEZETNÉL

C nyelv (kezdő)	40 óra	24 000 Ft
C nyelv (haladó)	20 óra	13 500 Ft
Turbó Pascal (kezdő)	50 óra	27 500 Ft
Turbó Pascal (haladó)	40 óra	24 000 Ft
PLC programozás	40 óra	32 500 Ft
Vision (kezdő)	15 óra	9 500 Ft
Vision üzemeltető	15 óra	9 500 Ft
Vision konfigurálás	50 óra	29 000 Ft
Üzemszervezés számítógépes támogatással	50 óra	26 500 Ft

Jelentkezés és további információ a következő telefonszámokon: 204-0172, 209-1039, 06-30-548-429

INFORMÁCIÓKÉRÉS: 0919 ▼

A http:// egyre divatosabb. A hp nem megy ki a divatból.

HEWLETT® PACKARD

Hivatalos viszonteladó

HP hálózati szoftverek, irodai PC-k, monitorok, nyomtatók, plotterek és kellékeik, scannerek, kalkulátorok, modemek árusítása

PC-alapú számítógépes hálózatok tervezése, kivitelezése és üzemeltetése, átalánydíjas szervizszolgáltatás

DOS, OS/2, NOVELL, UNIX és XENIX rendszerek telepítése. MICROSOFT, COMPUTER ASSOCIATES, COREL szoftverek forgalmazása megrendelés szerint

SONY

Audio/video eszközök kijelölt márkaboltja

ALINOR

ALINOR Kereskedelmi és Szolgáltató Kft.
1025 Budapest, Csévi u. 7.
Telefon: 393-1050
Telefax: 393-1055

Nyitva tartás: hétfőtől péntekig 10–18 óráig

INFORMÁCIÓKÉRÉS: 0901 ▲

PLUSZ PROFI 2000

Számítástechnikai szerviz Kft.



- Számítógép (PC) és nyomtató javítása, átalakítása, kiszállásos javítása
- Tápegységjavítás
- Szünetmentes áramforrások javítása
- Floppy- és CD drive-ok javítása
- Garancia megváltásos javítások
- Vírus detektálás és irtás



új átvevőhelyünk:

1111 Budapest, Irinyi J.u. 39. fsz. (ANSAN)

1047 Bp., Mildenberger u. 1/b.	☎ 180-4698
1054 Budapest, Vadász u. 19.	☎ 111-5456
1042 Budapest IV., Király u. 25.	☎ 379-4719
1111 Budapest, Irinyi J. u. 39. fsz.	**☎ 267-6025**

Szükség esetén cserekészüléket biztosítunk!

INFORMÁCIÓKÉRÉS: 0926 ▲

MCAFEE

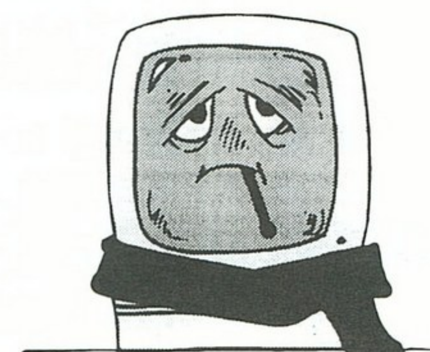
Network Security & Management

Hatékony és elterjedt vírusvédelmi rendszerek

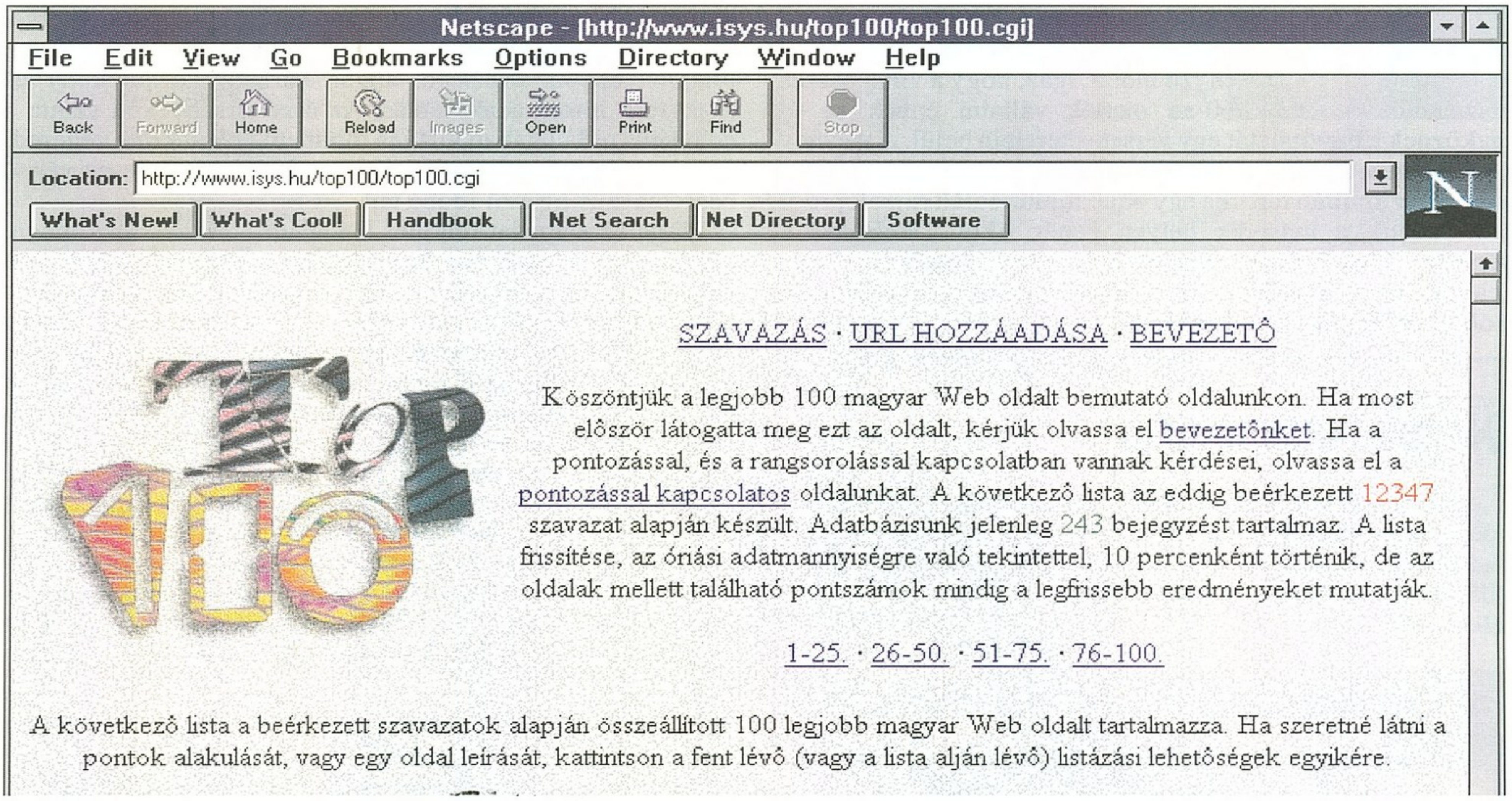
VirusScan	MUNKAÁLLOMÁS	**Windows (3.1, 95 és NT), DOS, OS/2, Solaris, Macintosh**
NetShield	SZERVER	**NetWare 3.x, 4.x, Windows NT**
BootShield	BOOT	**Boot Sector DOS, Windows**
WebScan	INTERNET	**E-mail / Internet Browser (Mosaic, Netscape ...)**

MCAFEE Inc. *hivatalos* képviselete:

PiK-SYS® Szolgáltató és Tanácsadó Kft.
H-1213 Budapest, Szentmiklósi út 18.
Tel.: (36-1) 276-0864
Fax: (36-1) 276-1235
Internet: piksys@mail.matav.hu



INFORMÁCIÓKÉRÉS: 0925 ▲



„Csak egy linkgyűjtemény”

Az Internet magyar szekciójának rendszeres látogatói egyre gyakrabban fedezhetnek fel olyan dolgokat, amelyek a hálózatot mint számítástechnikai eszközt állítják a középpontba. Ilyen volt a közelmúltban a Medián online közvéleménykutatása, és ilyen az iSYS által kezdeményezett Hungarian TOP 100-as lista is.

Számos helyre „drótozták be” azt az elektronikus szavazócédulát, amelynek segítségével a játékra hajlandó háló-látogatók arra adhatták le szavazataikat, hogy szerintük jelenleg melyik honlap a legjobb. Az értékelési szempontok között egyaránt szerepelnek tartalmi és formai ismérvek. Az előbbiek között pedig némiképpen pejoratív éllel az is, hogy az adott web-oldal valóban hasznos-e, újszerű-e, érdekes-e, vagy „csak” egy ugrópont-gyűjtemény. Pedig, ha úgy tetszik, a Hungarian TOP-100 sem más, „csak” egy linkgyűjtemény... No de milyen! A lelőhelyek olyan koncentrátuma, amely jelzi, milyen irányokban milyen fejlesztésekkel jelentkeznek a honi műhelyek. Az általános tájékozódás szempontjából tehát szinte nélkülözhetetlen ez a gyűjtemény.

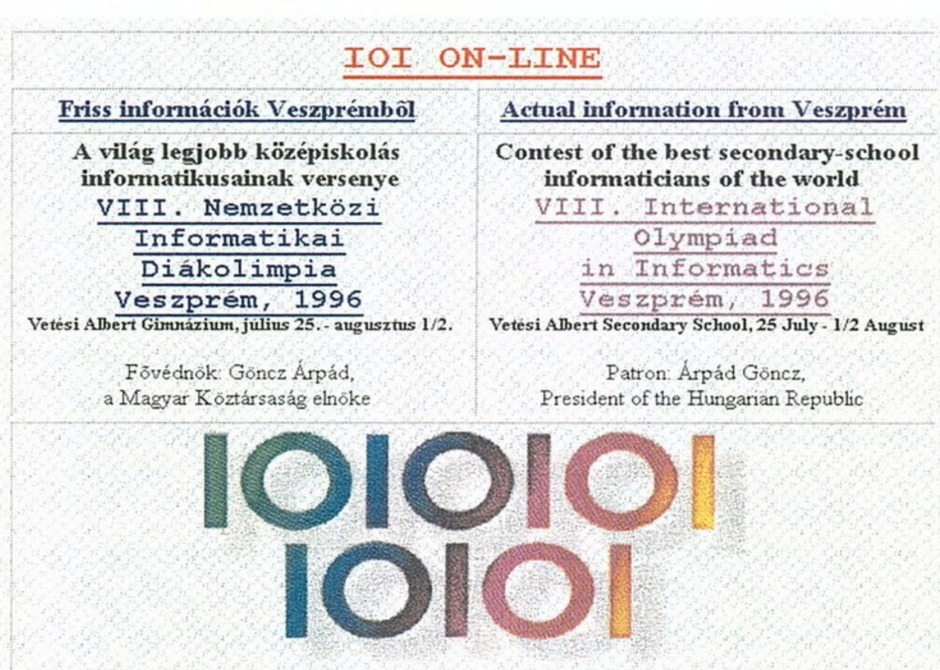
Nézzük a listavezetőket 1996. augusztus 4-5-én. Az élen az „Egyelőre TOP 100” c. slágerlista állt, de egy napra rá már az Internetto vette át az első helyet. Hasonló módon cserebérélt a Budapest OnLine és Feszt Timea Diana Éva is a 3-4. helyen. A listára kerültek között szép számmal szerepelnek a különböző médiakísérletek, kulturális, oktatási és politikai honlapok, és viszonylag szerény számban egyes számítástechnikai cégek hirdetőtáblái.

Program értékel programot

Az idén Magyarországon rendezett (8.) informatikai diákolimpiának egyik legérdekesebb oldala az értékelés volt, ugyanis minden programozási megoldás elemzését egy erre a célre kidolgozott program végezte el. Ritkán szembesítenek emberi alkotással szemben olyan bírát, amely nem mérlegeli az árnyalatokat, mércéje szerint valami vagy jó, vagy nem jó. Nem érdekli, hogy elegáns-e vagy szép-e a megoldás, csakis a produktum egzakt ismérvei alapján minősíti a végeredményt.

Bizony, milyen jó lett volna, ha valami hasonló program vizsgáztatta volna piacra kerülése előtt mondjuk, a CorelDraw-t vagy a sok Windows-változatot! (Bár nem biztos, hogy a piacra dobást a teszt eredménye hátráltatta volna, arról ugyanis már mindenképpen emberek döntenek.)

Egyébként a versenyen a 20 legjobb versenyző érdemelt aranyérmet (idén nem volt közöttük magyar), további 36-an kaptak ezüstérmet (egy magyar: Gosztolya Gábor), s még 52-en örülhettek a dobogó képzeletbeli harmadik fokának (ilyen érem két magyar versenyzőnek jutott: Lakatos Rolandnak és Tóth Lászlónak). Egyénileg egy cseh versenyző végzett a lista abszolút első helyén, mindössze 4 ponttal elmaradva a maximálisan megszerezhető 200 ponttól, a legjobb csapat pedig a kínaiaké volt.



Ugyancsak a veszprémi informatikai olimpiához kapcsolódik az első magyarországi Java-verseny eredményhirdetése is. Itt mindössze 8 versenyző indult, igaz, hogy a világ összes országából is csak 360-an merték vállalni ennek az új eszköznek a használatát egy verseny keretein belül. A verseny első helyezettje Paller Gábor „Graffiti" c. programja lett (online, szimultán falfirka egy objektumorientált rajzolóprogram révén), a második helyen Fehér Ákos „WeBasic"-e végzett (Basic-szerű, interpretált nyelvi környezet) míg „ca4D" nevű általános grafikus keretrendszerével Bajzát Róbert érdemelte ki a harmadik helyet.

Olimpiai informatika

Amikor nyilvánosságot látott a hír, hogy az IBM gondoskodik az atlantai játékok informatikai vérkeringésének biztosításáról, sokan „féltették" a Nagy Kéket — nehogy valami világra szóló blamázs kerekedjék ki a nagyformátumú vállalkozásból.



Ha a számos kritika ellenére végül is sikerült „megúszni" igazi rendszerösszeomlás vagy látványos fejreállás nélkül, ez nem jelenti azt, hogy ne lettek volna olyan informatikai fogyatékosságok, amelyek a jövőre nézve nagyon tanulságosak. Az olimpiai informatikát ugyanis nem lehet „rutinból" levezényelni, s utólag sem könnyű megmondani, hogy hol volt „lyukas" a rendszer, nemhogy előre felkészülni. Az olimpia mint műfaj olyan szintű rendszerintegrációs, rendszerszervezési feladat, amely mindenképpen igényli, hogy a fővállalkozó cég — még egy olyan nagyságrendű is, mint a szóban forgó — *saját* erejét „meghaladóan" is, professzionális partnerek nélkül is teljesíteni tudja.

Hogy mire gondolunk? Ha a sportágak szakszövetségei nem képesek prezentálni azt a tudásbázist, amelyet az informatikai rendszer alapul vesz (mondjuk, a mi Kőbán Ritánkról információ az aranyérem után is „not available"), akkor saját szervezői módszerekkel kell azt pótolni. Illetve, ha a rendszerintegrátor nem képes megmondani, mire is lenne tulajdonképpen szüksége, aligha várható gördülékeny és szakszerű információáramlás. Az úszóversenyeken szinte a célbaérés pillanatában világgá röpíthető volt a *teljes* végeredmény, a kajak-kenu szakágnál viszont még az érmesek végső besorolása is hosszú időbe telt.

Maga az alapadat-feltöltés bizonyult — legalábbis innen, távolból — a rendszer leggyengébb pontjának: vagy volt információ, de érdektelen, vagy pedig csak 0 bájtos állomány. Az olimpiát nézőnek szerintünk ugyanis nem olyan típusú információra van szüksége, hogy X. vagy Y. szereti a tulipánt és szívesen hallgat Mozartot, hanem arra, hogy pályafutása során hol, mikor, milyen versenyeken szerepelt, és ott milyen eredményeket ért el. Ha egy versenyzőről csak azt tudja a rendszer, hogy beszéli saját anyanyelvét, ez nem tekinthető *információnak*. (Közvetítéseik során cikizték is eleget ezt a fajta „adatszolgáltatást" mind a rádió-, mind a tévériporterek.)

Végül is az IBM elérte marketing-célkitűzéseit. Ha csendben, minden részletében kifogástalanul oldják meg az összes feladatot, talán még kevesebbet is beszéltek volna róla, mint így, a néhány bosszantó hiba kapcsán. Lehet, hogy reklámfogás volt?

Fotólabor helyett HP DeskJet 690C

Megszoktuk már, hogy egy-egy technológiai újítás kicsit átrendezi a nyomtatási módok használatának arányait. Ilyennek ígérkezik a Hewlett-Packard új színes tintasugaras nyomtatási eljárása is, amelynek 1996. augusztus 26-val bejelentett gépcsaládja rövidesen a fotóiparnak támaszthat igen komoly konkurenciát. A HP DeskJet 69xC sorozat tagjai a fotókhoz hasonló, ragyogó színű, részlet- és tónusgazdag képeket produkálnak, leginkább két jelentős újításnak köszönhetően. Az egyik a most először alkalmazott HP PhotoREt (photographic resolution enhancement technology), a másik egy új festékkazetta, a HP Photo Cartridge. Ez utóbbi a nyomtató normál festékkazettája mellett, egyszerűen a fekete festékkazetta helyére téve „besegít" a pasztellszínek és a finom tónusok nyomtatásánál. Különösen hatásos a nyomat fotóminősége, ha az erre a célra ajánlott új papírt használják hozzá. (Alábbi illusztrációnk forrása is egy ilyen kép volt.) Figyelemreméltó az ár: egy szokásos méretű (10x15 cm-es) kép kinyomtatása már indulásnál is csak mintegy 30 forintba kerül, tehát veszedelmesen megközelíti a hagyományos negatív filmről készült fényképek kópiánkénti árát.





Dante Poklában az elkárhozottaknak nyakig az izzó lávában kellett állniuk. Nem mintha a mai üzleti világban ilyen szörnyűek lennének a dolgok – végül is az Ön rendelkezésére áll a GroupWise™, a Novell® „groupware" megoldása. Az „Universal In Box" lehetővé teszi, hogy megszervezze E-mailjeit, találkozóit, faxait, sőt a hangpostáját is – s mindezt egy helyről. Ha kedve úgy tartja, üzenetei kezelésére használhatja laptopját, telefonját, vagy személyhívóját, bárhol is van a földön (vagy a Pokolban). Lépjen hát a több mint ötmillió felhasználó nyomába, akik már megtalálták a megoldást! *Ha meg akarja tudni, hogyan maradnak más cégek a csúcson, hívja a 266-7770-es telefonszámot vagy látogasson el a http://www.novell.com internet címünkre.*

INFORMÁCIÓKÉRÉS: 0